Optimal Mesh Topology Generation for CFD

Zaib Ali*

School of Mechanical and Manufacturing Engineering, National University of Sciences and Technology, H-12, Islamabad, Pakistan.

Paul G. Tucker

Department of Engineering, University of Cambridge, Cambridge, CB2 1PX, United Kingdom.

Shahrokh Shahpar

CFD Methods, DSE, Rolls-Royce, plc., Derby, England DE24 8BJ, United Kingdom.

Abstract

Multiblock structured mesh generation is amongst the most popular meshing techniques for flow simulations. However, the time consuming and user intensive process of blocking complex domains presents a major bottleneck in structured mesh generation. To this end, we have considered various automatic blocking strategies where some typical aeroengine domains have been showcased. An adjoint based error analysis procedure is then used to assess these methods. It is found that, in general, the medial axis based approaches provide optimal blocking and yields better accuracy in computing the functional of interest. This is because the medial axis based methods produce meshes which have better flow alignment especially in case of internal flows. A new hybrid blocking approach, combining the existing methods with the distance field isosurface is also presented to overcome the shortcomings of the current methods. Such a blocking technique has the potential to be applied to a wide range of flow domains.

Keywords:
Multi-Block Structured Meshing, Medial Axis Transform, Automatic

*Corresponding author
Email address: zaibali@gmail.com (Zaib Ali)

Blocking, Distance Field, Adjoint Based Error Analysis.

1. Introduction

1.1. Structured Quad/Hex Meshing

Computational fluid dynamics (CFD) has now become an essential part of the modern day design hierarchy. Each step in the CFD design process from the CAD definition to the optimum design needs to be as computationally efficient and reliable as possible. Mesh generation has remained one of the most time consuming steps in the CFD process and faces even bigger challenges with the ever increasing need for simulating complex three dimensional (3D) flows. There is a trade-off between the mesh quality, ease of generation, solver requirements and parallel mesh generation when choosing amongst the structured and unstructured mesh types. Unstructured meshes offer more flexibility for meshing complex 3D domains. Structured meshes, on the other hand, offer higher numerical accuracy and less storage than the unstructured meshes. They also allow relatively easy implementation of high-order finite volume and finite difference schemes.

The impact of a particular mesh type on the solution accuracy can be substantial. This has been illustrated, for example, in [1], where the case of modelling the spatial development of a subcritical Tollmien-Schlichting (T-S) instability wave in a plane channel is considered. The hexahedral and the triangulated meshes are used here. The mesh snapshots for a lower channel half, that contains the T-S wave, are shown on the right hand side of the Figure 1(a) and 1(b). The curves in Figure 1 show the decay of vertical velocity fluctuation $v^{'}$ for the subcritical T-S wave. Figure 1(a) compares the performance of both meshes for a second order, finite volume, cell vertex solver while Figure 1(b) shows the same comparison for a cell centered solver. The analytic solution to the Orr-Sommerfeld equation is represented by symbols. The poor performance (too diffusive) of the triangulated mesh is clear from both graphs while the hexahedral mesh provides a reasonable match with the analytical solution. It is also noted that this triangulated mesh results in an approximately 30% increase in the number of edges, thus raising the computational cost. Hence, where feasible, the use of structured quad/hex meshes can be advantageous to the unstructured tetrahedral meshes.

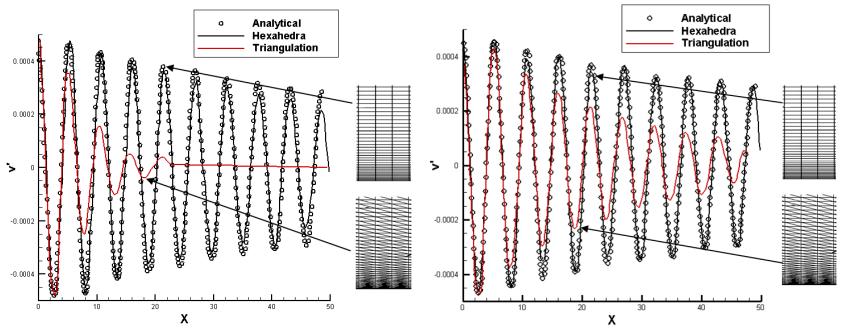


Figure 1: Plots of T-S wave decay with streamwise distance for different mesh types, (a) cell vertex solver (b) cell centred solver, produced from [1].

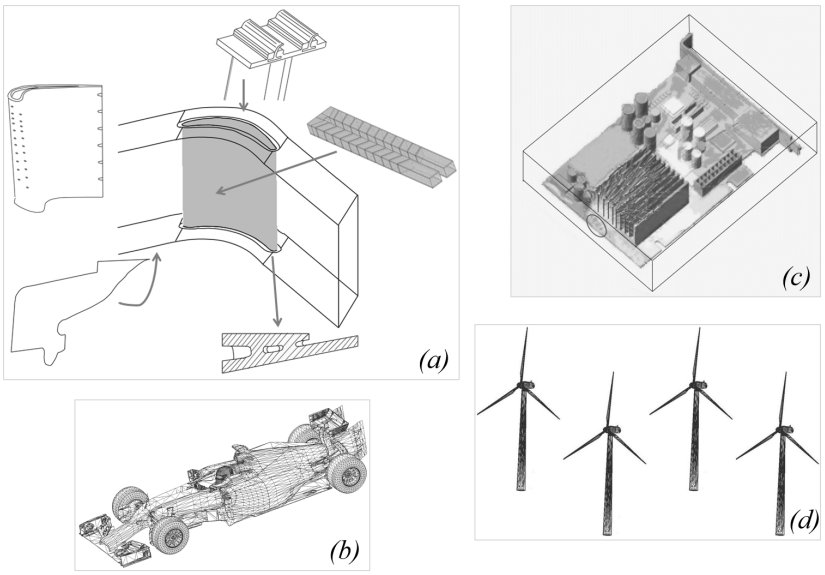


Figure 2: Meshing challenges: (a) Turbine blade (counter-clockwise from top left) cooling holes, seal cavity, cutback trailing edges, internal cooling passages, shroud cavity, centre: standard H-O-H blocking for the blade passage [2, 3]. (b) a racing car geometry for aerodynamics analysis (c) electronics systems for thermal and ventilation analysis (d) a wind farm domain for efficient wind turbine placement study.

1.2. Multiblock Topology Generation

Multiblock structured mesh generation is among the most widely used meshing techniques in flow simulations. This is essentially a two-stage process. In the first stage, a suitable blocking topology is generated which divides the complex domain into simple sub-domains. The resulting blocks are subsequently meshed. This structured blocking offers an efficient meshing strategy for topologically simple configurations like the main gas flow path and standard templates exist for partitioning of such domains. For example, the H-O-H type blocking is commonly used to mesh the blade passage as shown in the center of the Figure 2a [2, 3]. However, the modern day design challenges demand the computational analysis of more realistic geometries. A turbine blade, for example, has cooling holes, internal cooling passages, cut back trailing edges, shroud cavities and rim seals as depicted in the Figure 2a. Examples of other complex flow domains include, for instance, a racing car geometry for aerodynamics analysis, an electronics system for thermal and ventilation analysis and a wind farm domain for an efficient wind turbine placement study, as shown in frames (b), (c) and (d) of the Figure 2 respectively. Meshing such multiply linked and more diverse geometries requires significant user intervention, or writing of templates as part of a library [3]. Thus, an automatic blocking strategy can be beneficial to reduce the CFD design cycle time and could be a better alternative to the unstructured or hybrid meshing methods.

Fully automatic 3D block topology generation is a complex problem and currently there is no ideal block topology generating algorithm with all the desired features for structured mesh generation. However various automatic blocking approaches (see Figure 3) have been proposed with varying levels of automation and geometric complexity handling.

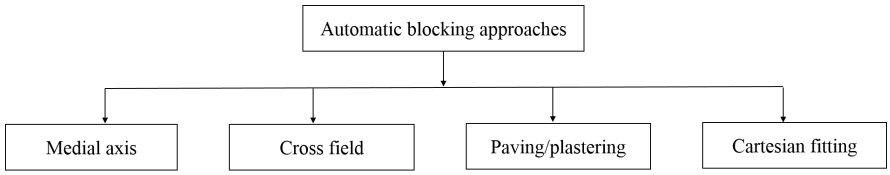


Figure 3: Automatic blocking approaches.

The medial axis transform (MAT) based algorithms offer an efficient approach towards automatic blocking. The medial axis, first proposed by Blum [4], can be defined as the set of lines and curves (surfaces in 3D) produced

from the midpoint of a maximal inscribed disc (sphere in 3D) as it rolls around the domain as shown in the Figure 4. The exact computation of the medial axis is difficult due to the underlying algebraic complexity. Also, the medial axis is unstable because a small perturbation in the domain boundary can result in a large change in the topology of the medial axis. Therefore, various approaches have been proposed to approximate the medial axis. These include surface sampling approaches [5, 6, 7, 8], thinning algorithms [9, 10, 11, 12], distance field based algorithms [13, 14, 15, 16] and the algebraic methods [17, 18, 19, 20, 21].

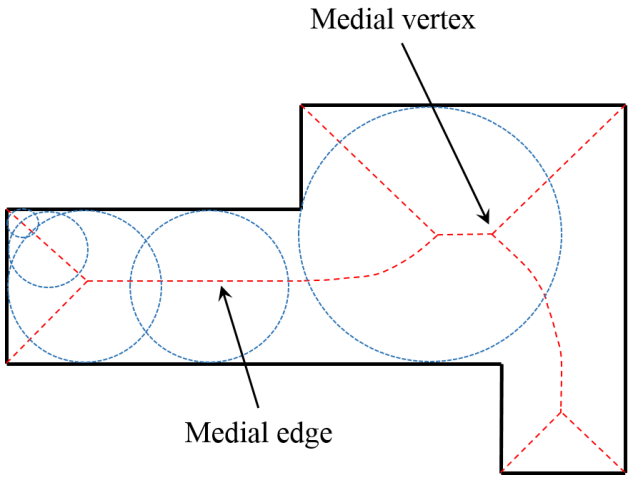


Figure 4: Medial axis of a domain.

The Voronoi diagram and the distance field based methods are more widely used for block topology generation. Voronoi diagram based methods such as those presented by Armstrong and collaborators [22, 23, 24, 25] use the constrained Delaunay triangulation of points distributed on the boundary to assemble 'shape molecules' featuring a mesh topology. The shape molecules are reduced to 'shape atoms' by inserting cuts between the medial vertices which topologically represent 4-, 5- and 6-sided polygons. A subdivision is created resulting in one block for each medial vertex, medial edge and medial face. A midpoint subdivision is then used for meshing the blocks. The concave features are removed by splitting the edges which are chosen based on a corner angle criteria. An alternative has been presented by Rigby [26], called the 'TopMaker' approach, which makes use of medial vertices and parts of medial axis to block the domain. Medial vertices are defined as the

points which are equidistant from three locations form the domain boundary. Consequently, six types of medial edges and appropriate rules are defined for creating the blocks. Further enhancements have been included to produce a good quality mesh however this technique has yet to be extended for 3D.

LayTracks3D [27], is a hybrid hexahedral meshing method combining medial axis based decomposition and the advancing front method. Here, the medial axis is used directly to create the mesh instead of the domain decomposition. Regularly spaced thin strips are created between the medial radii pairs from the medial axis to the associated touch points. A further subdivision is then carried out to form 'Tracks'. The advancing front method is then applied from the boundary towards the medial axis to generate the mesh. This technique produces good quality hexahedral meshes but degenerate cells can be formed around the sharp concave features.

The use of the normal wall distance is widespread ranging from turbulence modeling and mesh generation in CFD to computer vision [28, 29, 30, 31, 32]. The distance field can be evaluated through search procedures, integral based methods and by solving differential equations like eikonal equation [30, 29]. Distance field based approaches are also widely used for the medial axis approximation. The 2D Euclidean distance map based approach has been used in [13] where an analysis of the local directional maxima is carried out to obtain the medial axis. An enhancement to this technique for higher dimensions has been presented in [14]. A nearest-neighbor search based criteria to perform a spatial subdivision is used in [15] for approximating the medial axis. Another hybrid approach called differential MAT or d-MAT approach is presented in Xia and Tucker [16]. The hyperbolic-natured eikonal, level set equation is used to calculate the distance field [33]. Medial axis point clouds are then extracted from the Laplacian or Hessian determinant of the distance field. A thinning algorithm is then used for thinning the point clouds into curves and surfaces. The method is illustrated in the Figure 5(a-c) for a simple domain. Such a hybrid approach thus avoids complexity of pure geometric approach and provides more accuracy than the pure image thinning algorithms. A further enhancement to this approach is a biased MAT proposed in Xia and Tucker [34]. This adds more flexibility to the subsequent domain decomposition as the biased medial axes give tighter blocks surrounding the object and more blocks between objects. The differential equation based distance field and subsequent medial axis approximations have the advantage that a customized medial axis can be computed by manipulating the boundary conditions. This is shown in Figure 5(d) where the distance field

relative to the part having the wall boundary condition is computed. This results in the simpler medial axis represented by the red curves in the Figure 5(f) thus avoiding triangular blocks encountered in, for example, Figure 5(c). In addition to the collision of wave fronts forming the compression shocks i.e. the medial axis, the expansion waves shown in the Figure 5(e) can be utilized to aid the blocking process. The expansion features are connected to the nearest 'medial vertex'; that is if that point lies within sight of the fan caused by the expansion feature. If not, it is connected to the nearest point on the shock feature topology(see green lines as Rule 1 in Figure 5(f)). Rule 2 is borrowed from the TopMaker method [26]; Hanging features (features which are not connected to anything) are extended to the nearest point on the geometry. This is shown by the green line in Figure 5(g).

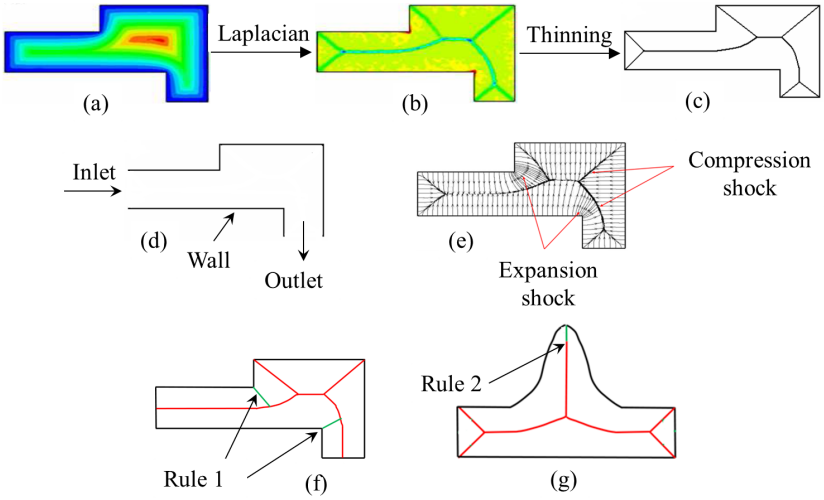


Figure 5: d-MAT method and blocking; (a) distance field of a simple domain (b) medial axis extraction through the distance field Laplacian (c) point cloud thinning (d) simplified medial axis (e) shock and expansion features of the wave fronts (f) blocking rule 1 (g) blocking rule 2.

Recent advancements in mesh generation are the methods based on the cross-fields (frame fields in 3D). A cross field is defined by assigning a set of four unit vectors to points at the discrete locations. These unit vectors form a regular cross on the tangent plane. Thus the size and the orientation of the quadrilateral cells can be specified by the cross field. A number of approaches have been put forward for 2D and 3D cross field based domain decomposition and mesh generation. To generate the block topology, the partitioning

created by connecting the cross-field streamlines to the singularities can be used. The resulting blocks of the cross field can then be mapped to a grid. Such methods have been presented in, for example, [35, 36, 37, 38].

Fogg et al. [39] also use the cross field for block topology generation. Their method is similar to the paving algorithm. The cross field is initialized at the boundaries and then propagated inwards using the fast marching method. A continuous cross field is obtained up to the location where the advancing fronts collide giving rise to singularities and the resulting multi-block decomposition. Such a method is simpler than the paving algorithm because the mesh connectivity is not catered for in the cross field. A variation of this method is presented in [40] where a medial axis based blocking technique with implicit use of the cross field is described. This results in effective handling of the concavities and sharp features as compared to the pure medial axis based techniques such as TopMaker.

Kowalski et al. [38] use a PDE based approach to obtain the cross field. A diffusion problem is solved for this purpose. After locating the field singularities, the lines in the cross field connecting the singularities are extracted that result in the domain partitioning. Huang et al. [41] and Li et al. [42] use energy function minimization to smooth the frame fields. However, here a rough initialization of the surface cross field is used which can result in poor singularity locations and hence can deter the generation of all hexahedral meshes. Kowalski et al. [43] describe a frame field method that does not require an initial surface cross field. Other approaches towards cross field based mesh generation are described in [44, 45, 46]. The cross field approach towards domain decomposition and mesh generation is novel and efficient but quite complex and expensive.

Malcevic [47] presents another automated blocking strategy based on a Cartesian fitting method. While preserving the topology definition, a forward geometry simplification is performed followed by fitting the model into a Cartesian framework. The next step is blocking the domain after which the blocked model is mapped back on to the original geometry. Further operations such as removing singularities by J-grid wrapping are performed to enhance the mesh quality. This technique has been applied for meshing the end-wall cavities found in turbomachinery. This technique is very simple and but has only been demonstrated for 2D cases so far. The method sometimes produces some unnecessary mesh clustering across the block interfaces.

1.3. Error Estimation

Several approaches have been previously presented in the context of error estimation. A review of these approaches is presented by Roy [48]. The error estimators can be divided into two main categories: solution based and adjoint based. The solution based error indicators are based on the flow gradients or undivided differences to adapt the certain flow features like shock waves, wakes, boundary layers and slip lines etc [49, 50, 51, 52]. However, adaptation based on such indicators might not guarantee a reduction in the overall error estimate and can lead to erroneous results [53, 54, 55, 52].

The solution based error estimators are based on local measures of error. However, the other parts of the domain might have a strong influence on that local error estimate. Hence as stated above, the global error might not reduce by treating the local error. This is the case, for example, in convection dominated flows, where the errors upstream may affect the solution downstream. This is described in [56, 54], where an accurate capturing of the shock position through the local pressure gradient based error indicator was desired. The continuous local refinement based on this error estimate resulted in a shock position different to the one obtained by the uniform global refinement. This is because the error upstream of the shocks are convected and affect the strength and position of the shock. Thus engineering quantities of interest such as lift and drag become highly sensitive to the discretization and to the residual errors propagated from elsewhere in the flow field. In contrast the adjoint based error estimators can efficiently quantify the impact of such errors and cater for the propagation effects. The error estimate thus obtained can be used for accurate prediction of quantities of interest and subsequent mesh adaptation.

One of the earliest demonstrations of the adjoint error estimation and mesh adaptation is presented by Müller and Giles [57] in which the dominant part of error correction term is expressed as the global sum of the dot product of the adjoint solution and the residual error. Building upon the technique described in Giles and Pierce [58] and Müller and Giles [57] , Venditti and Darmofal [55] use the additional concept of a 'truth' mesh to estimate the error in the objective function. A truth mesh is simply a successive uniform refinement of the working coarse mesh. Error in the coarse-mesh functional is estimated with respect to its value on the fine mesh by prolongation of the coarse-mesh non-linear (also called primal) and the adjoint solution on the fine grid. Hence the solution on the finer grid is not required. This strategy provides improved error estimates than the one presented in [57].

This approach has been extended to 3D by Park [59]. One of the difficulties associated with the approach presented by Venditti and Darmofal is the storage of the fine mesh which is an additional memory overhead. An alternate approach has been presented in Dwight [60] which still employs the adjoint method but the sensitivity of the objective function with respect to the added stabilizing dissipation in the numerical method is used as an error indicator. While this alleviates the need for storage of the fine mesh, the disadvantage of this strategy is that sources of error other than the added numerical dissipation are not included in the error indicator. The approaches presented in Fidkowski [61] and Nemec and Aftosmis [62] demonstrate the use of the Venditti technique for Cartesian meshes. Mani and Mavriplis [63] present the adjoint based error estimation and adaptation for unsteady flows. A detailed review on the topic of error estimation using adjoint methods is presented by Fidkowski and Darmofal [64].

The current work compares the d-MAT [16], TopMaker [26], cross field [39] and the Cartesian fitting [47] based block topologies by using the adjoint based error analysis. The use of this adjoint application is intended to inform template and algorithm design for multi-block meshes. Hence, the prohibitive cost and complexity of the adjoint method is not an issue. A hybrid approach combining the distance field contours and the Cartesian fitting for mesh generation is also proposed. Although the cases considered here a specific to aeroengine aerodynamics, the methods used and the techniques developed are general and can be utilized across a range of flow domains.

2. Adjoint error estimation

2.1. Discrete Adjoint Analysis

After the primal flow solution is available, the discrete adjoint equations are solved to get the adjoint variables. This section presents the discrete adjoint analysis. A detailed derivation can be found in [65]. Let Q, the flow variables at discrete set of points with coordinates X, be the solution of system of steady non-linear equations

$$R(Q, X, \alpha) = 0 \tag{1}$$

where R is the discrete residual vector and α is a set of design variables. Also consider an objective function $J(Q, \alpha)$ which one wishes to optimize. The sensitivity of this objective function to a set of design variables can be

expressed in the following form

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial Q}\frac{\partial Q}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \tag{2}$$

The adjoint variables ν can be defined as the effect of the flow residual on the objective function:

$$\nu = \frac{\partial J}{\partial R} \tag{3}$$

Using Equations (1) and (3), Eqn. (2) can be written as

$$\frac{dJ}{d\alpha} = -\nu^T \frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \tag{4}$$

Eqn. (4) can be re-written to give the following equation for adjoint variables

$$\left(\frac{\partial R}{\partial Q}\right)^T \nu = \left(\frac{\partial J}{\partial Q}\right)^T \tag{5}$$

This set of linear equations can be solved to give the adjoint flow variables in a manner similar to the primary flow solution.

2.2. Adjoint Error Analysis

An adjoint error estimation procedure similar to that of Venditti and Darmofal [66] is followed in this work. A detailed description of this method is presented in the Appendix. The global error in the objective function $J(Q)$ can be related to the local residual error with the adjoint variables working as the weighting function (see Appendix). This relation is described by the equation

$$J\left(Q\right) - J_h\left(Q_h^H\right) = -(\nu_h^H)^T R_h\left(Q_h^H\right) \tag{6}$$

where H and h represent the coarse and the fine mesh levels respectively.

2.3. Flow Solvers

Simulations in this work are performed using the Rolls-Royce in-house HYDRA program which is a coupled suite of non-linear, linear and adjoint CFD solvers [67]. The core non-linear solver has an edge based data structure and the flow equations are integrated around median dual control volumes using a MUSCL based flux differencing scheme [68]. It uses a block Jacobi

pre-conditioner and iteration towards the steady state is carried out using a 5-stage Runge-Kutta scheme. For accelerated convergence, an element-collapsing multigrid algorithm is used. All the HYDRA solvers are fully parallel and uses OPLUS library for MPI communication.

HYDRA Adjoint, the steady adjoint solver, is based upon the discrete adjoint approach [69]. The gradients are evaluated using the automatic differentiation. First, the non-linear primal flow version is linearized and then the adjoint version is obtained by transposing the linearized equations. This methodology has successfully been used in design of turbomachinery components [70].

3. Results

In this section, various automatic blocking methods are compared using three different quasi 3D/3D geometries. These blocking techniques are based on variety of approaches like medial axis, cross field and Cartesian fitting. In addition to the automated blocking methods, the results from the manually created block topologies are also included for comparison. These hand-crafted block topologies were independently generated prior to the start of this current work.

The adjoint error estimation procedure is employed for the assessment. The objective functions chosen for all the cases is the total pressure loss (as percentage of total inlet pressure) and the outlet capacity defined as $\dot{m}\sqrt{T_0}/P_0$. Here, $\dot{m}, T_0$ and P_0 are the mass flow rate, total temperature and total pressure respectively at the outlet. The error is computed with respect to a fine mesh which is a two-step uniform refinement of the original coarse mesh.

In the implementation of the d-MAT approach [71], curve fitting is used to generate the medial axis from the point cloud instead of the pixel thinning. The distance field is computed using the search procedure algorithm available in the HYDRA flow solver. To create a valid block topology from the MAT, the d-MAT approach uses additional rules (see Figure 5). The TopMaker [72] and the Cartesian fitting [73] techniques have been manually followed for these cases. The cross field block topologies are generated in the Rolls-Royce mesh generator program called Abihex [74, 39]. The meshes are generated in the commercial mesh generator Pointwise [75].

3.1. Labyrinth seal

Labyrinth seals are used in aero-engines to reduce the flow leakage. The medial axis transform (MAT) of the domain is shown in the Figure 6(a).

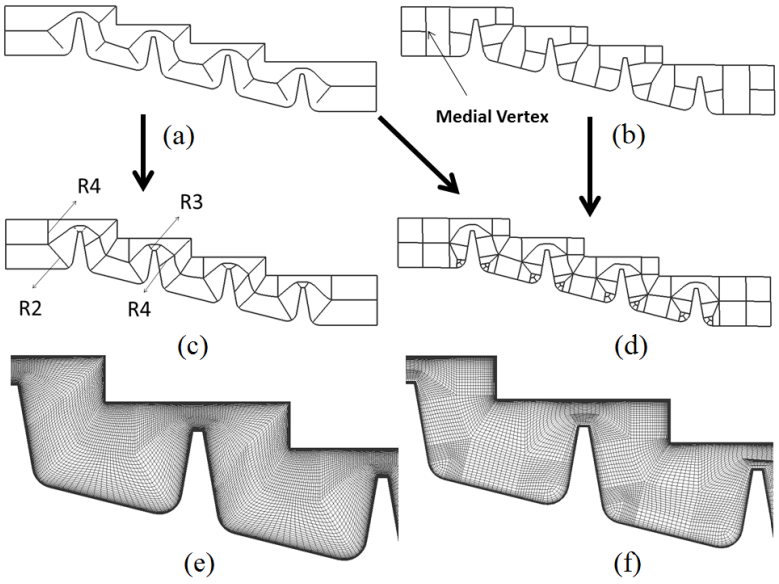


Figure 6: The block topologies and resulting meshes for the labyrinth seal using the medial axis based approaches (a) medial axis (b) medial vertices with their associated touch points (c) d-MAT blocking with rules (d) Topmaker blocking (e) d-MAT mesh (f) TopMaker mesh.

The blocking layout using the d-MAT approach with rules is shown in the Figure 6(c). The blocking following the TopMaker approach is shown in the Figure 6(b). The meshes generated from the d-MAT and the TopMaker topologies are shown in the Figure 6(e) and Figure 6(f). The d-MAT approach produces blocks resulting in a mesh which is aligned with the flow direction, especially around the seal teeth. The TopMaker blocking is similar to the d-MAT except that there are additional small blocks around the round sections.

The Cartesian fitting method tries to align the block topology in the Cartesian frame. To enhance the mesh quality, an additional O-grid wrap (also called the J-wrap) is also used around the rounded features of the domain as shown on the left of Figure 7(a). As can be seen in the figure, the resulting mesh has more orthogonality than the other two approaches. However the effect of cell clustering to resolve the boundary layers extends

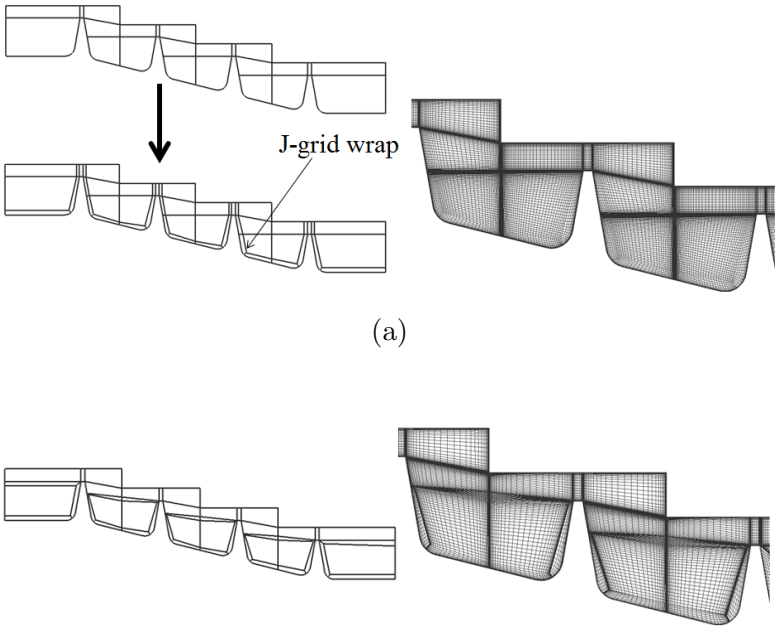


Figure 7: The block topologies and resulting meshes for the labyrinth seal (a) Cartesian fitting (b) Manual

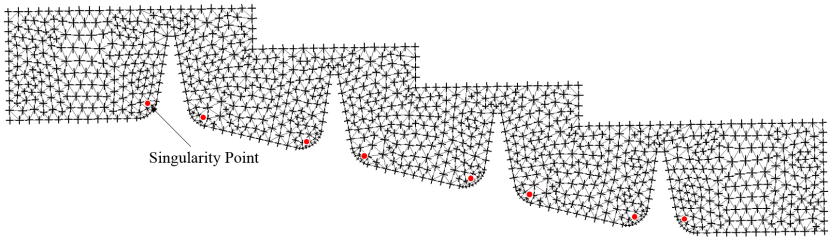
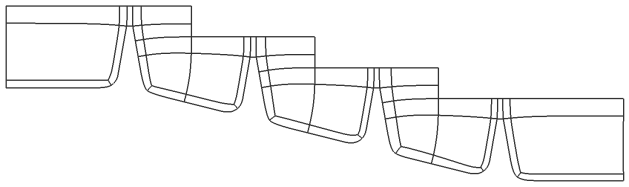


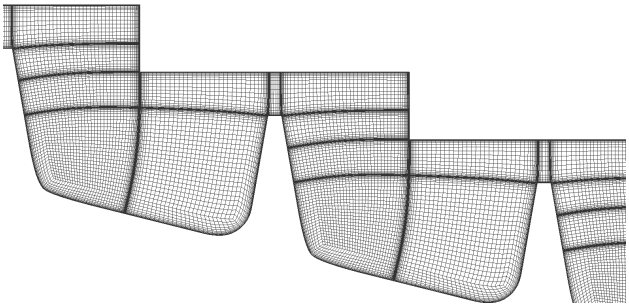
Figure 8: The cross field with singularity locations on the labyrinth seal domain.

out into the main flow path. This is necessary to retain the mesh quality across the block boundaries. Interestingly, the hand-crafted blocking (Figure 7(b)) looks similar to the one generated by the Cartesian fitting method.

The cross field with singularity locations (red dots) for the seal domain is shown in the Figures 8. The cross field blocking and the resulting mesh for the labyrinth seal case are shown in the Figure 9. As can be observed, the result has a strong resemblance to the Cartesian Fitting and the manual



(a)



(b)

Figure 9: Labyrinth seal (a) cross field blocking (b) cross field mesh.

meshes.

The simulations are performed using the Spalart-Allmaras [76] (SA) turbulence model with an intensity of 10% at the inlet. The flow is modelled with an axial Reynolds number of 10,000 based on average seal radius. The upper part of the cavity is stationary while the lower part rotates at $\Omega = 200\pi$ rad/s. The mean rotor radius is $R_{ave} = 0.253m$. The axial velocity contours of the primal flow and its adjoint counterpart for d-MAT mesh are shown the Figure 10(a)-(c). Regions of flow acceleration past the seal tooth tips and flow recirculation at the bottom of the steps are generated in the primal flow field. The adjoint flow field shows the sensitivity of the objective function to the residuals in the axial velocity. Both the positive and negative values of the adjoint variables are important, when seen in the context of design optimization. For example, the positive values of the velocity adjoint variable indicate a shape change that induces an increase in the momentum flux and vice versa.

The error in computing the objective function (total pressure loss and capacity) is estimated using the method described above. The overall error

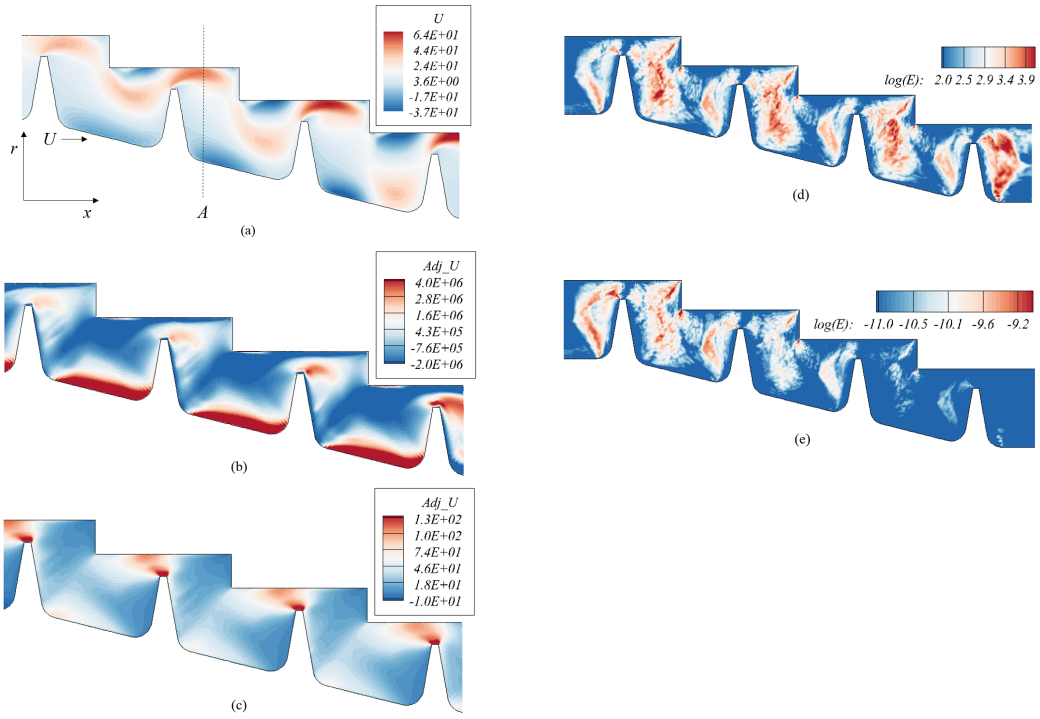


Figure 10: Labyrinth Seal: the contours of (a) axial velocity (b) adjoint counterpart of the axial velocity (objective function: total pressure loss) (c) adjoint counterpart of the axial velocity (objective function: capacity) (d) error map (objective function: total pressure loss) (e) error map (objective function: capacity).

Blocking Type	N(Approx)	TE_N (P_{loss})	TE_N (Capacity)
d-MAT	37,000	1.0	1.0
TopMaker	36,000	1.1	1.1
Cross Fields	39,000	1.2	1.4
Cart.Fitting	41,000	1.3	1.5
Manual blocking	42,000	1.5	1.7

Table 1: Labyrinth Seal: coarse mesh cell count and the total normalized adjoint based error.

can then be obtained by summing individual error contributions. If E_j is the estimated error, then the total weighted error TE over the whole domain for

N number of cells is given by

$$TE = \frac{\sum_{j=0}^{N}(|E_j| \times Vol_j)}{\sum_{j=0}^{N} Vol_j} \tag{7}$$

where Vol is the cell volume (area in 2D). A comparison of the error estimates can then be made by normalizing the total error with the d-MAT error TE_{dMAT} such that the normalized value TE_N is given by

$$TE_N = \frac{TE}{TE_{dMAT}} \tag{8}$$

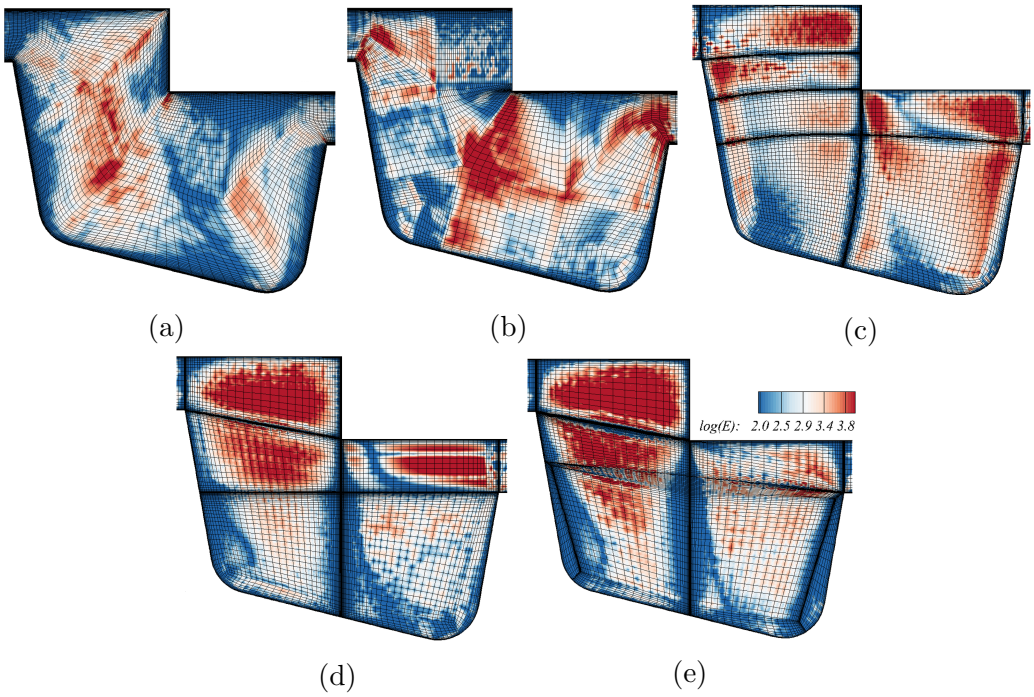


Figure 11: The error maps (objective function: total pressure loss) for the 2nd step of the seal (a) d-MAT (b) Topmaker (c) Cross Fields (d) Cartesian fitting (e) Manual.

Figure 10(d) and Figure 10(e) show the error maps for the two objective functions. As can be seen, these maps are different to each other. The regions identified by the adjoint error indicator, in the total pressure loss case, are same in the every inner step of the seal. On the other hand, in the

capacity objective function case, the error indicator shows high sensitivity in the steps closer to the inlet. This illustrates the usefulness of the adjoint weighted error estimation where, despite having the same residual error in both cases, only areas critical to the optimization of the particular objective function are marked.

Table 1 shows the coarse mesh cell count N and the normalized error estimates TE_N for all block topologies. The results indicate that the medial axis based meshes provide the most accurate value of the objective function than the other block topologies. One of the reasons for this is that for the similar cell count, the medial axis based approach generates a mesh which has better alignment with the flow and also a more uniform cell size distribution in the main flow path than the other approaches. Figure 11 highlights this point where the enlarged view of second seal step is shown. In order to keep the mesh sizes consistent between the blocks with boundary layer refinement and their neighbors, the cross field and the Cartesian fitting meshes are clustered in the main flow path. This constraint effects the overall uniformity of the mesh size producing coarser mesh in the areas away from the block boundaries, thus contributing to the error. Similar reasoning applies to the manually generated mesh.

Figure 12 shows the histogram of the cell skewness for all the block topologies. When looking at the mesh quality from the viewpoint of the geometrical mesh quality metrics such as skewness, the d-MAT mesh performs badly despite providing the more accurate solution. This shows a lack of connection between the solution accuracy and the traditional quality metrics implying that the quality is mainly dependent on the physical solution. Nevertheless, the mesh quality can have a strong influence on the solution convergence and can be used to improve defective meshes.

The error convergence plots of the meshes generated by the various techniques for both objective functions are shown in the Figure 13. These plots show faster convergence for the medial axis based approaches. Despite the fact that when highly refined, meshes based on any blocking may yield the desired level of accuracy, the medial axis based meshes can be advantageous for simulations where it is too costly to use a highly refined mesh, for example, when performing design optimization.

3.2. Rim seal

The second geometry used in this section is a stationary and axis-symmetric rim seal. The geometry, the block topologies and the meshes generated by

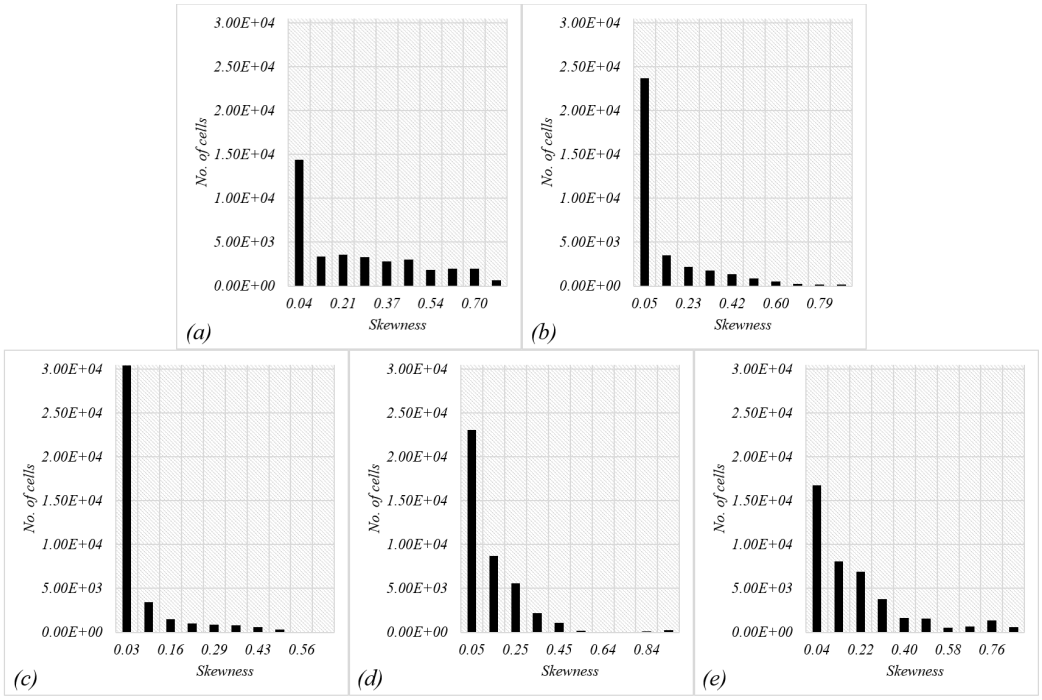


Figure 12: Labyrinth seal: Skewness histogram (a) d-MAT (b) Topmaker (c) Cross Fields (d) Cartesian fitting (e) Manual.

applying the d-MAT, TopMaker, cross fields, Cartesian fitting and the hand-crafted blocking are shown in the Figure 14. The medial axis and cross field block topologies have similar features despite a significant difference between the number of blocks. The manual blocking in this case produces more uniform mesh and less number of blocks than the rest of the methods. The simulations are performed using the SA turbulence model. The contours of the Spalart-Allmaras variable (SA variable) and its adjoint counterpart for the d-MAT mesh are shown in the Figure 15. The strong sensitivity of the adjoint field near the inlet indicates how an inlet design change upstream might influence the separation downstream.

Table 2 compares the coarse mesh cell count N and the normalized error estimates TE_N for various approaches. As depicted by the statistics, the manually generated mesh produces more accurate functional estimates than the rest of the methods. This is followed by the d-MAT mesh. The results are similar for both objective functions. Figures 16 and 17 shows the error maps and the mesh alignment with the flow for various approaches. The

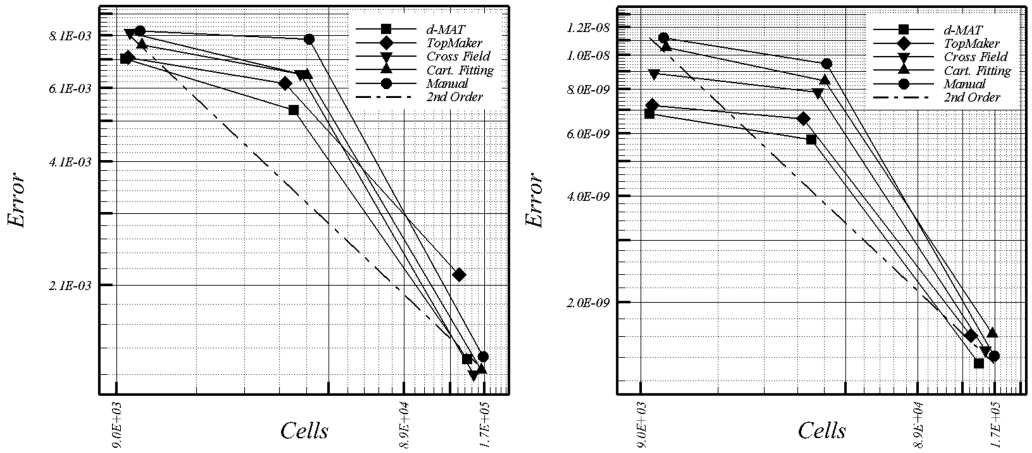


Figure 13: Labyrinth seal: error convergence (a) (objective function: total pressure loss) (b) (objective function: capacity).

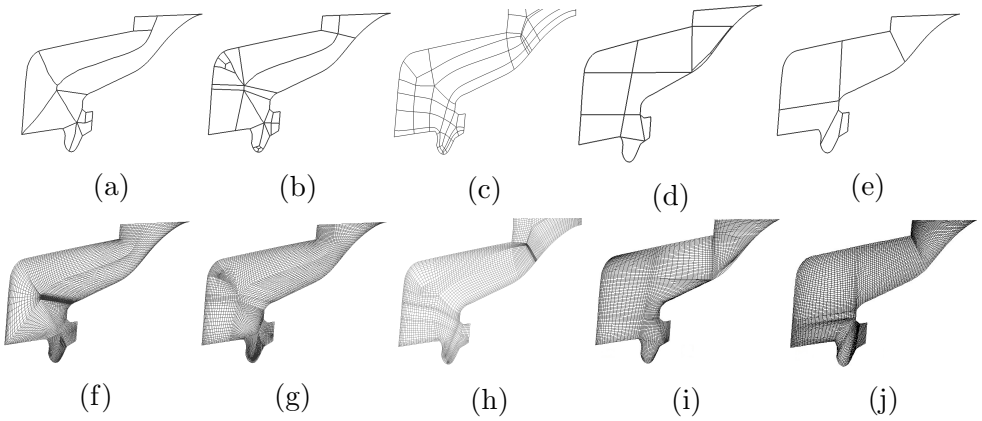
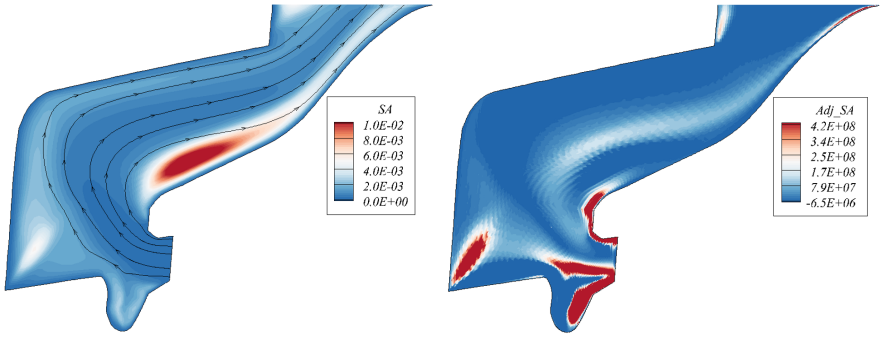
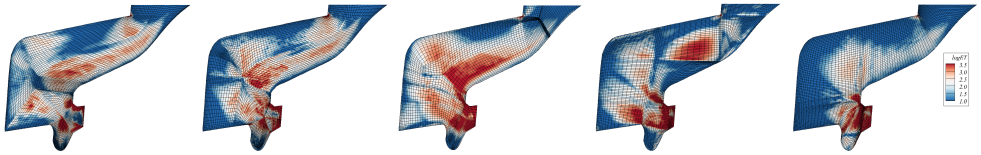


Figure 14: Rim seal: (a) d-MAT blocking (b) Topmaker blocking (c) Cross Fields blocking (d) Cartesian fitting blocking (e) Manual blocking (f) d-MAT mesh (g) Topmaker mesh (h) Cross field mesh (i) Cartesian fitting mesh (j) Manual mesh.



(a) (b)

Figure 15: Rim seal: contours of (a) Spalart variable (b) Adjoint counterpart of the Spalart variable for the d-MAT mesh.



(a) (b) (c) (d) (e)

Figure 16: The error maps (objective function: total pressure loss) for the rim seal: (a) d-mat (b) Topmaker (c) Cross field (d) Cartesian fitting (e) Manual.

Blocking Type	N(Approx)	TE_N (P_{loss})	TE_N (Capacity)
d-MAT	4,900	1.0	1.0
TopMaker	4,900	1.1	0.9
Cross Fields	4,800	1.3	1.2
Cart.Fitting	4,600	1.9	2.0
Manual blocking	4,900	0.8	0.7

Table 2: Rim seal: coarse mesh cell count and the total normalized error.

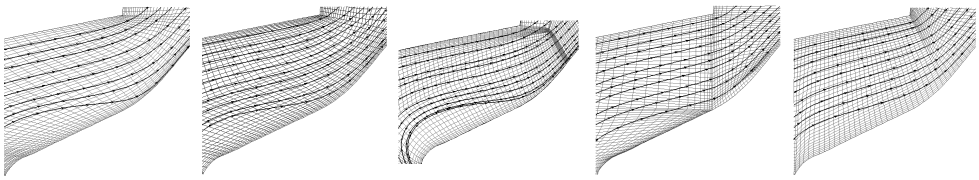


Figure 17: Rim seal: streamlines against the mesh (a) d-mat (b) Topmaker (c) Cross field (d) Cartesian fitting (e) Manual.

manually generated mesh performs well here due it smoothness and better flow alignment. The d-MAT, TopMaker and the cross field meshes are less aligned with the flow as compared to the manual mesh. Less degree of mesh smoothness can also be observed in these meshes. The Cartesian meshes also struggles to achieve uniform cell size distribution and the alignment with the flow thus resulting in the highest error. The cell skewness histograms and the error convergence plots for this case are shown in the Figure 18 and Figure 19 respectively. The convergence plots also show better performance of the manual and the medial axis based meshes.

3.3. Engine intake

In this section, a 90^o sector engine intake rig geometry as shown in the Figure 20 is used. This experimental setup has been used to study the intake lip flow in cross winds [77, 78].

The domain is partitioned using four different block topologies. For the 3D medial axis generation, the Voronoi diagram based algorithm of Dey and Zhao [7] is used. The d-MAT and TopMaker block topologies are then completed using additional rules as described in the Figure 5 (also see [26]). The Cartesian fitting blocking is manually followed. The hand crafted blocking used in [77] has also been included here for comparison. The block topologies and the resulting meshes are shown in the Figures 21 where a substantial difference in the blocking layout can be observed. The medial axis based meshes wrap around the intake lip geometry nicely. The Cartesian fitting mesh retains its typical nature along with mesh clustering extending away from the boundary layer. The cross field blocking used in the previous sections is not yet available for 3D cases and, hence, is not included here for the analysis. The coarse mesh cell count is given in the Table 4.

A range of simulations using this setup was carried out in [77] with varying

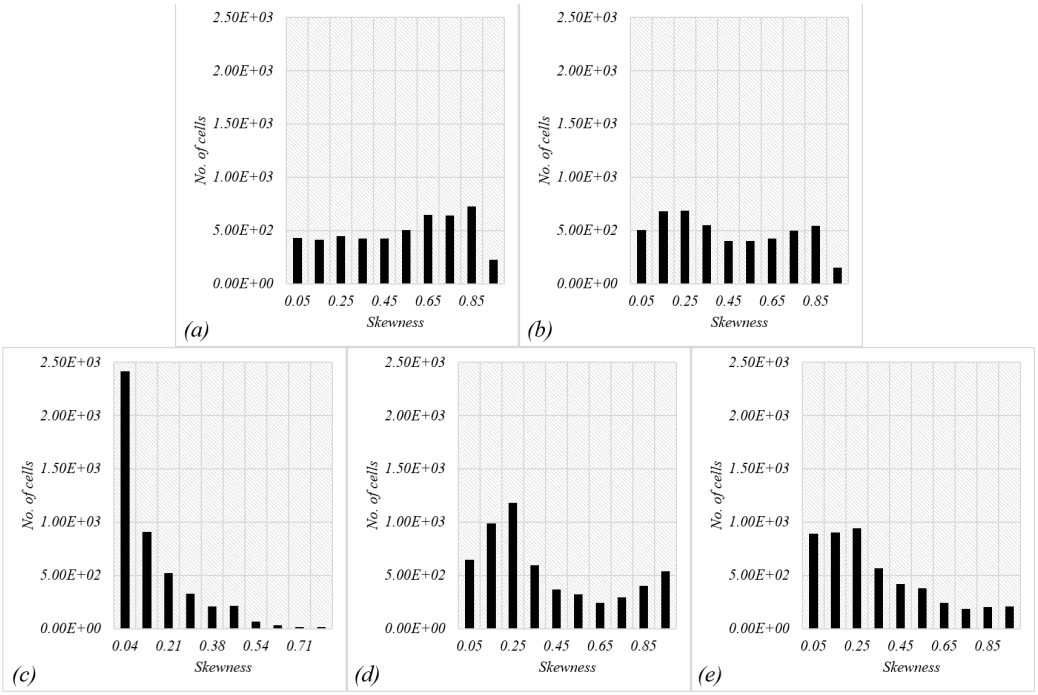


Figure 18: Rim seal: Skewness histogram (a) d-MAT (b) Topmaker (c) Cross Fields (d) Cartesian fitting (e) Manual.

exit Mach number (Ma_{ex}). The pre-separated and post-separated intake flows were investigated including the effect of surface roughness. Here a pre-separation case ($Ma_{ex} = 0.42$) is used. The operating conditions are given in the Table 3. The Reynolds number based on the exit diameter (Re_D) is approximately 7×10^5. The simulations are performed using the Spalart-Allmaras (SA) turbulence model, with near wall grid spacing giving $y^+ \approx 1$. The axial velocity contours around the intake lip, using the Roe's scheme, are shown in the Figure 22(a) along with the flow streamlines. The adjoint flow is then computed with the total pressure loss as the objective function. The contours of the adjoint Spalarat variable are displayed in the Figure 22(b) where a high adjoint sensitivity along the intake lip can be seen. In the context of the design optimization, these contours inform on how changing the lip design of the intake could improve the value of a particular objective function.

The adjoint error estimate results are given in the Table 4. It can be observed from table that the medial axis based meshes again outperform the

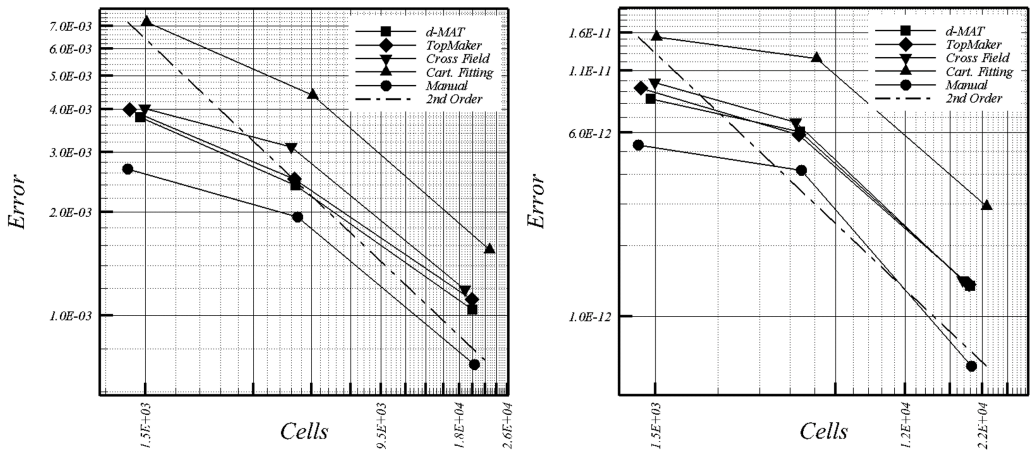


Figure 19: Rim seal: error convergence (a) (objective function: total pressure loss) (b) (objective function: capacity).

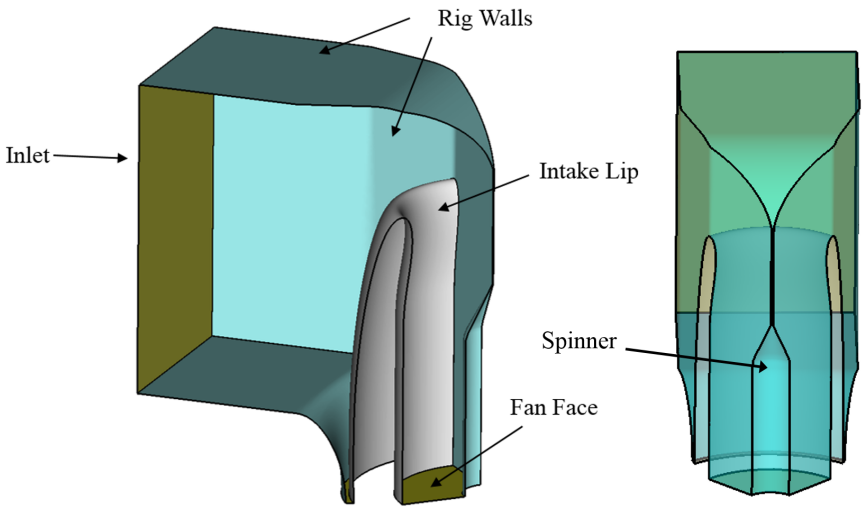


Figure 20: The 90^o sector intake rig geometry.

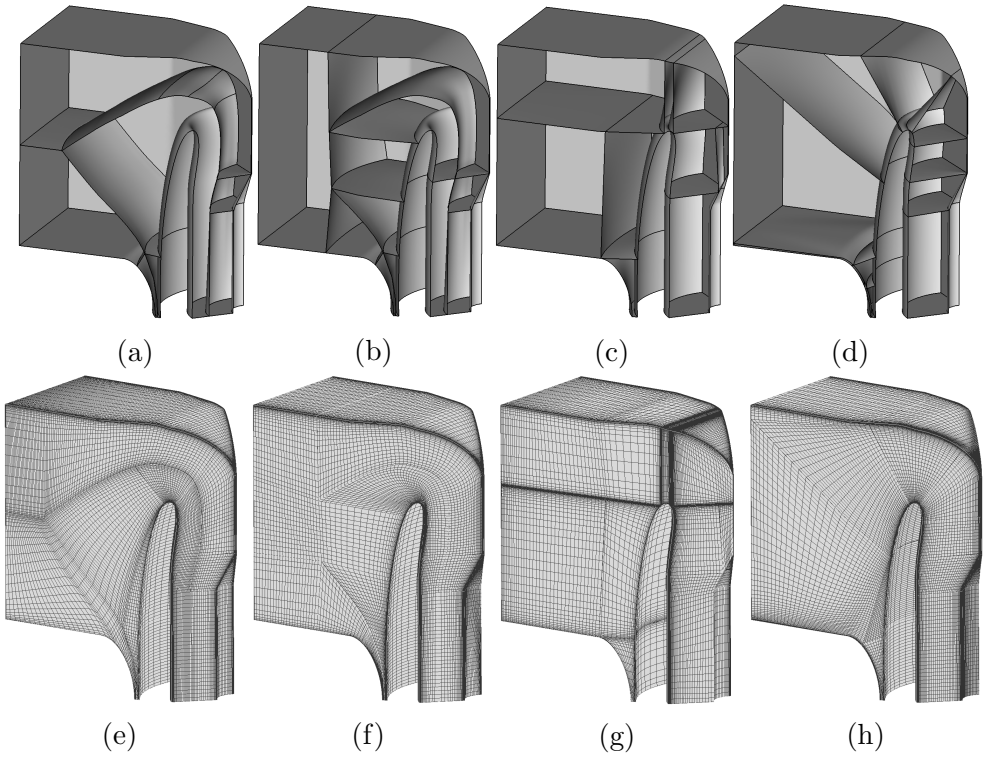
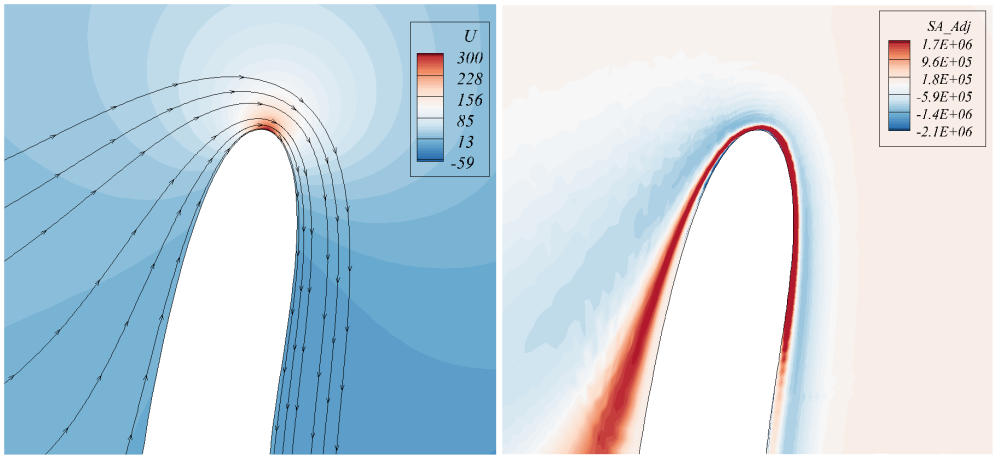


Figure 21: 3D engine intake rig: (a) d-MAT blocking (b) Topmaker blocking (c) Cartesian fitting blocking (d) Manual blocking (e) d-MAT mesh (f) Topmaker mesh (g) Cartesian fitting mesh (h) Manual mesh.

Exit Mach number	0.42
Inlet stagnation pressure	235073 Pa
Inlet static pressure	234763 Pa
Inlet temperature	314.07 K
Exit pressure	207996 Pa

Table 3: 3D engine intake rig operating conditions.



(a) (b)

Figure 22: 3D engine intake rig: contours of (a) Axial velocity (b) adjoint counterpart of the Spalart variable around the intake lip.

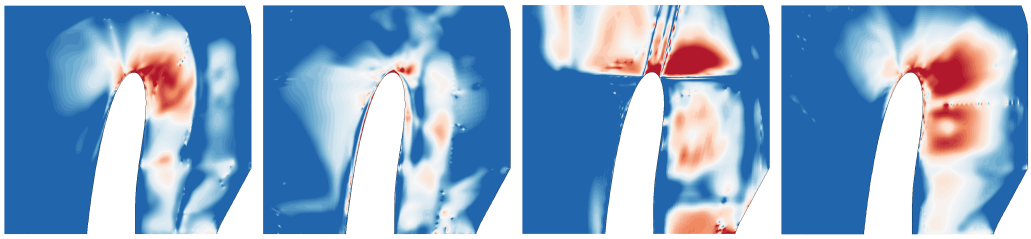
rest of the block topologies.

Blocking Type	N(Approx)	TE_N (Roe)
d-MAT	450,000	1.0
TopMaker	430,000	1.01
Cart.Fitting	450,000	1.5
Manual blocking	440,000	1.3

Table 4: 3D engine intake rig: coarse mesh cell count and the total normalized error.

The adjoint error maps for all the meshes are shown in the Figure 23. These are taken at z=0 slice where a consistent error pattern for both the schemes can be seen. The cell skewness histograms for various meshes are displayed in the Figure 24 which again show the disconnect between the

traditional quality metrics and the accuracy of the flow solution. The results are also compared with the experimental data as shown in the Figure 25. Here the Ma_{is} along the lip surface center line is plotted against the dimensionless lip length, L_D. This length is non-dimensionalized by the distance from the lip highlight to the fan face. A better agreement with the measurements can be seen for the medial axis based meshes.



(a) (b) (c) (d)

Figure 23: The error contours near the intake lip (a) d-MAT (b) TopMaker (c) Cartesian fitting (d) manual.

3.4. Hybrid Blocking

The previous section has demonstrated that the medial axis based blocking techniques provide an efficient domain partitioning option. However, handling complex 3D domains is still a challenge and medial axis based methods cannot always guarantee a quality block topology. For example, consider an aero-engine jet-wing-flap (JWF) domain with a far-field as shown in the Figure 26. Here, the medial axis close to the JWF geometry is unable to result in a quality block topology as shown by 2D slice of domain in the Figure 26. The solid lines represent the shock features i.e. the medial axis and the dashed lines show the expansion features.

To overcome this limitation, the distance field function d can be used. An isosurface (contour in 2D) of d is wrapped around the geometry to facilitate the MAT based block topology generation. The distance field computation is an essential step in the d-MAT approach and hence is available for use without any extra cost. This hybrid blocking procedure (see Figure 27) is described below with the help of a simple 2D JWF geomtery. The extension of this methodology to the 3D cases also follows the same procedure as demonstrated later in the section.

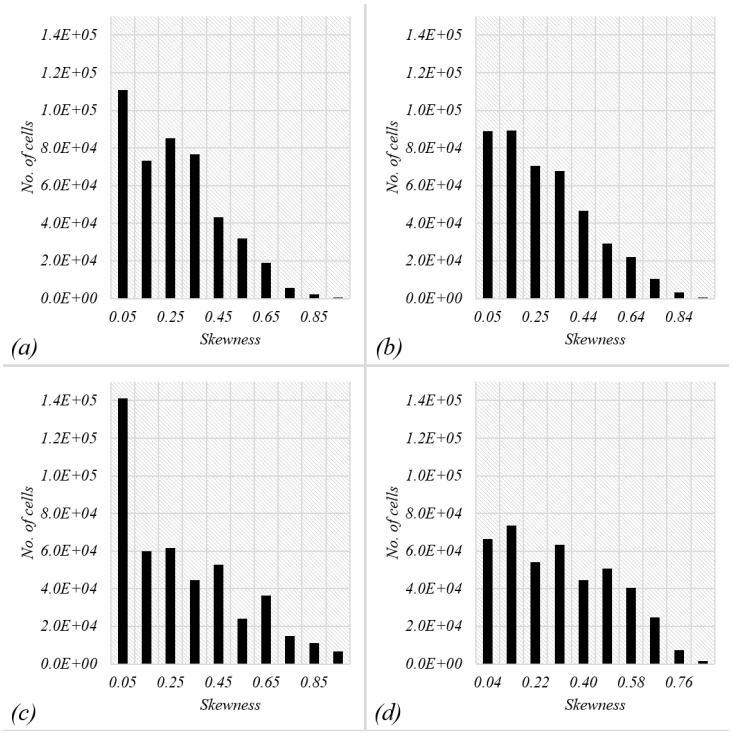


Figure 24: 3D engine intake: Skewness histogram (a) d-MAT (b) TopMaker (c) Cartesian fitting (d) manual.

- The distance field d is computed around the domain of interest as shown in the Figure 28(a). A suitable isosurface is then extracted from d. This isosurafce selection is currently arbitrary but it can be linked to a criteria. For example, it could be based upon the y^+ value. The isosurface acts like a virtual geometry or a wrap around the real domain (see 28(b)).

- The next step is approximation of the medial axis between the geometry and the distance field wrap. The Voronoi diagram based algorithm of Dey and Zhao [7] is used here for the medial axis approximation. This algorithm provides a more stable and continuous medial axis for complex 3D domains than the voxel thinning approach used in the d-MAT blocking method. The input to this program is the point cloud data of the geometry and the distance field isosurface. The medial axis is generated as a continuous surface which can be imported into the

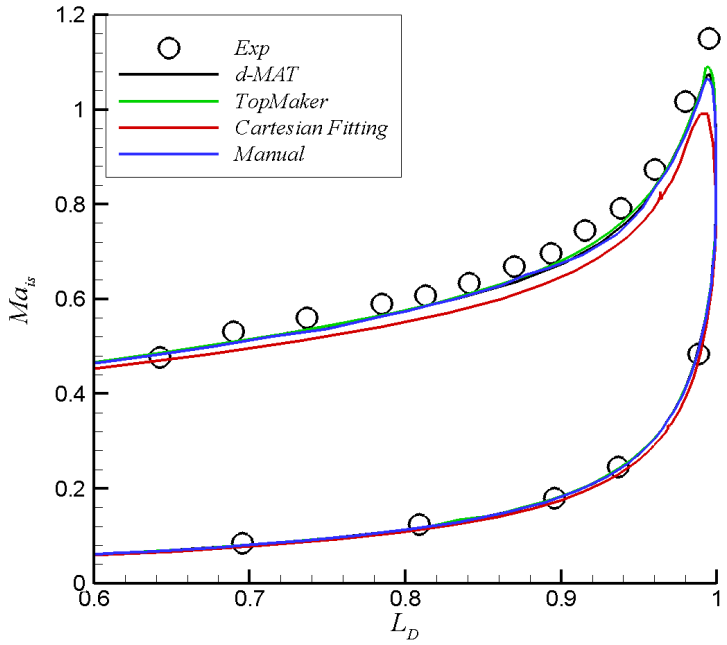


Figure 25: Lip isentropic Mach number profiles.

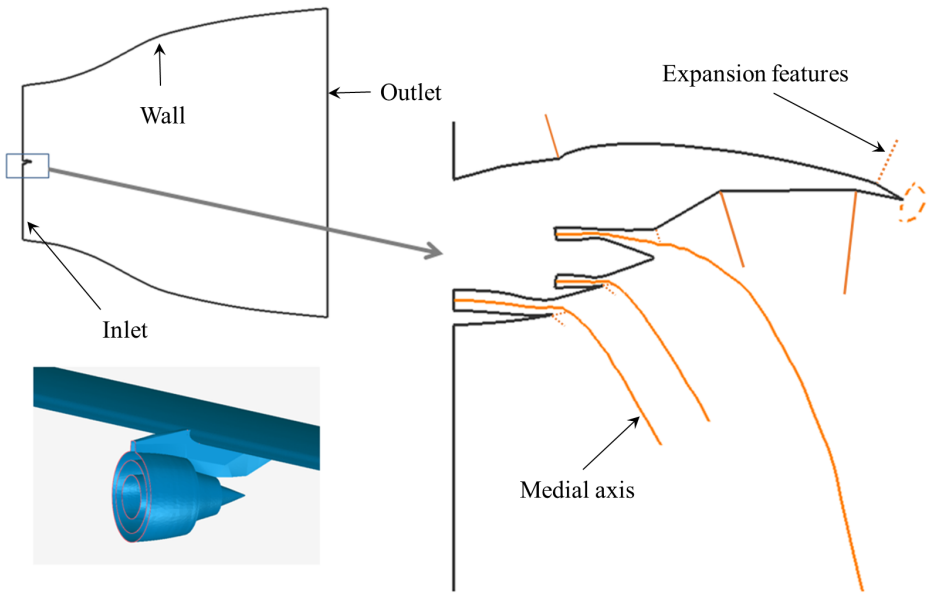


Figure 26: Medial axis transform for the jet-wing-flap close to the geometry.

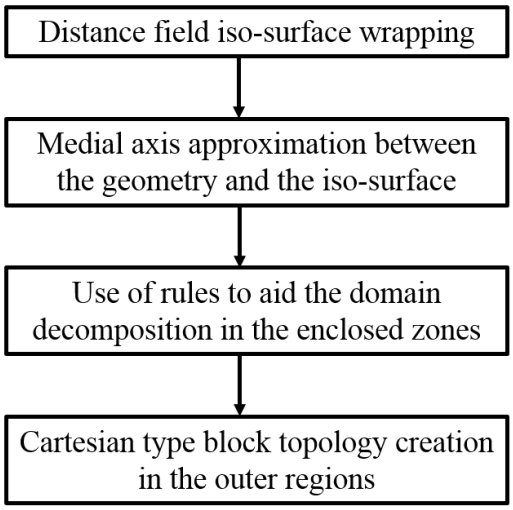


Figure 27: Hybrid blocking procedure.

mesh generator. The medial axis for the JWF slice is shown in the Figure 28(c).

- To complete the blocking process, rules described in the Figure 5 are used. For example, as shown in the Figure 29, the expansion features are connected to the nearest medial vertex or otherwise the medial axis.

- Once the critical parts of the domain have been blocked using the medial axis, the far-field region can be partitioned using simple Cartesian fitting or H-type blocks. This is shown, for example, in Figure 29 with the green lines. There can still be some regions where the block topology is still valid but not of good quality. Such areas must be manually altered. Hence, a semi-automatic blocking process arises.

3.4.1. NASA CRM wing-body-tail

In this section, the hybrid blocking is applied to partition the domain around a 3D NASA Common Research Model (CRM) horizontal wing-body-tail configuration. This model represents a modern, transonic and commercial aircraft designed to cruise at $M = 0.85$ and $C_L = 0.5$. The geometric and aerodynamic details about the model are described in [79, 80]. The configuration is shown in the Figure 30(a). The far-field is represented by a cube with boundaries ten fuselage lengths away upstream and downstream

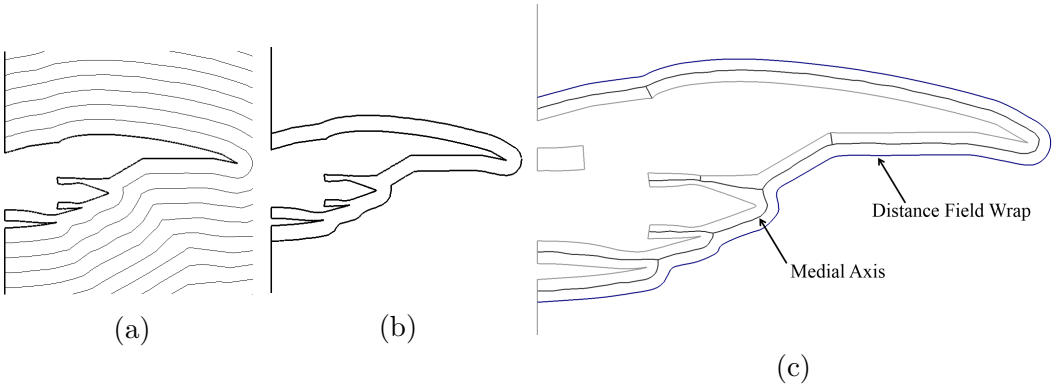


Figure 28: Two dimensional jet-wing-flap geometry: (a) the distance field; (b) distance field wrap and (c) corresponding medial axis.

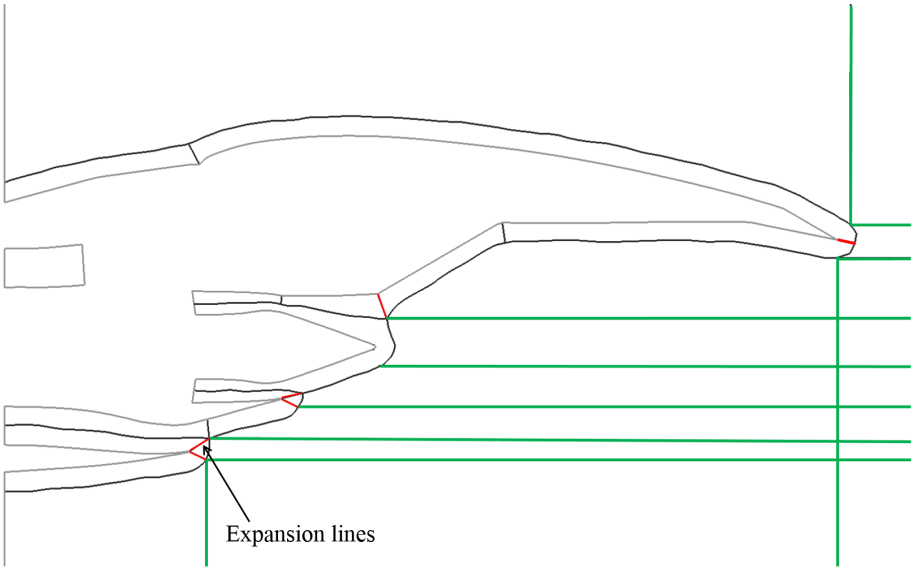
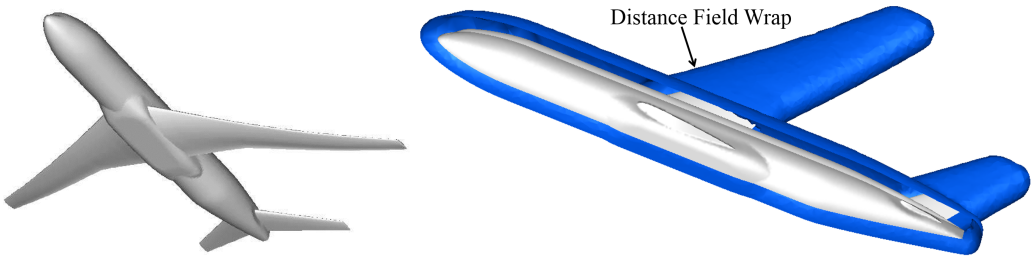


Figure 29: Hybrid blocking aroud the 2D JWF slice.

and ten spans away from the symmetry plane. First, the distance field wrap is computed around the model geometry which is shown in the Figure 30(b).

The medial axis is approximated around the model wrapped by the distance field isosurafce. A cut section of the model geometry and the medial axis is shown in the Figure 31. As can be seen, the medial axis branches at the wing-fuselage junction providing a useful block partitioning. The medial axis around the wing and tail also provides a block topology similar to O-



(a) (b)

Figure 30: NASA CRM: (a) wing-body-tail configuration (b) configuration with the distance field wrap.

type or C-type meshes. To assist the blocking, as shown in the Figure 32, expansion features at the trailing edges of the wing and the tail are joined to the nearest medial axis . After the blocking around the geometry is complete, the far-field domain partitioning is carried out. The region is partitioned to create a H-type mesh. The block topology around the model is shown in the Figure 33. The volume and the surface mesh cuts are displayed in the Figure 34. The cell skewness histogram is shown in the Figure 35 where approximately 85% of the cells have skewness value below 0.55.

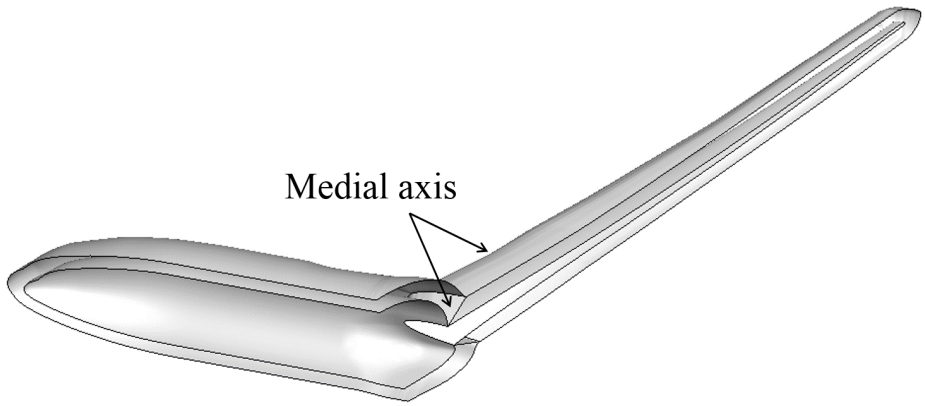


Figure 31: NASA CRM wing-body-tail cut section with the medial axis.

The NASA CRM configuration has been the test case for the 4th and 5th AIAA CFD drag prediction workshops [80, 81]. Here, we use the same flow conditions as given in the workshop to compute the flow around the test case. The simulation is carried out at $M = 0.85$ and $C_L = 0.5$ with Reynolds

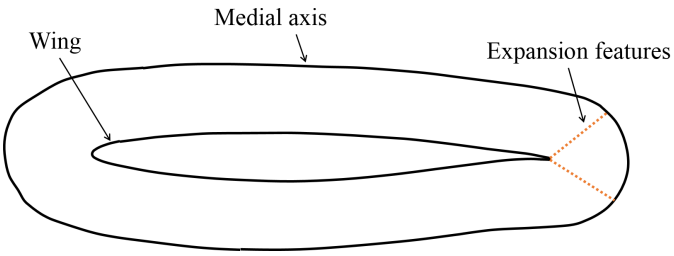


Figure 32: NASA CRM wing-body-tail blocking around the wing section.

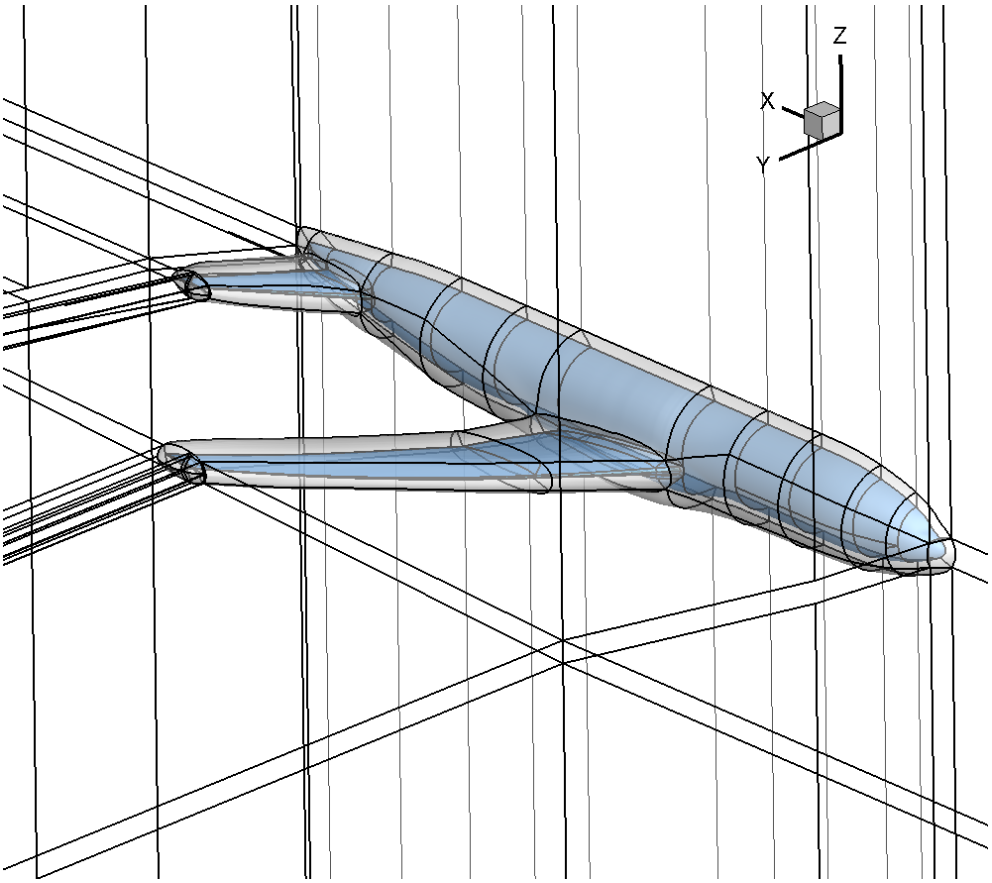


Figure 33: NASA CRM wing-body-tail hybrid blocking.

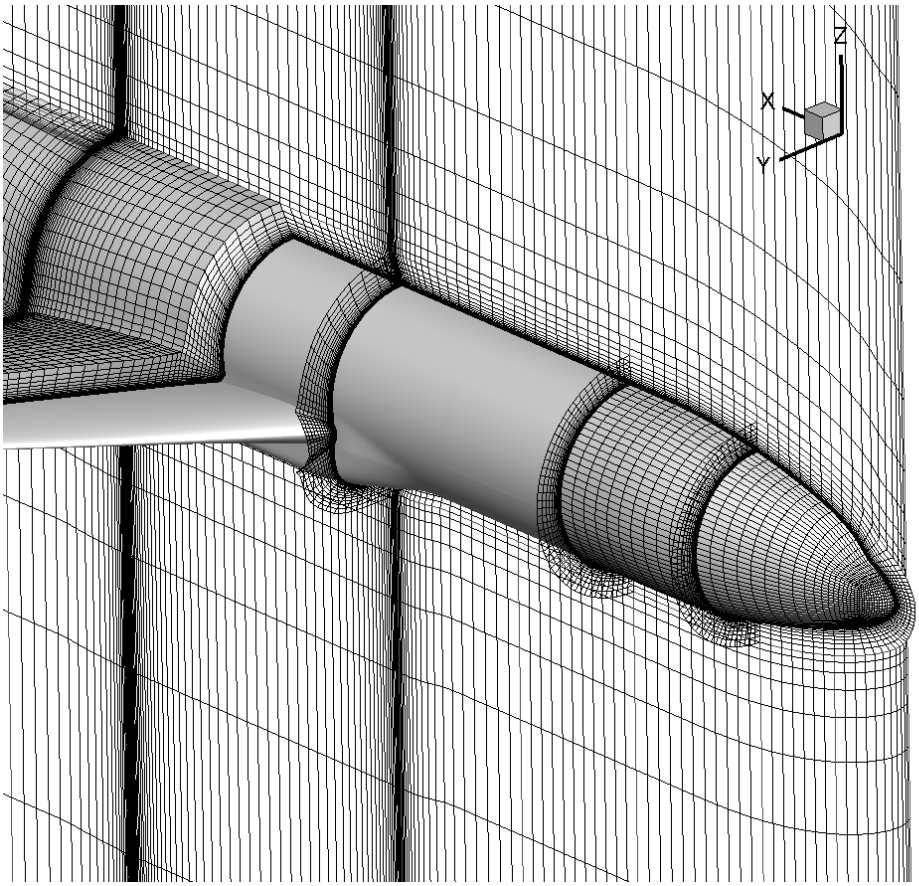


Figure 34: NASA CRM wing-body-tail mesh.

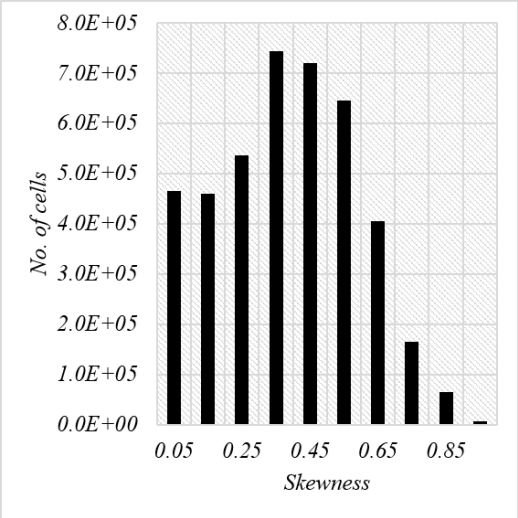


Figure 35: NASA CRM wing-body-tail cell skewness histogram.

number of 5×10^6 based on the reference chord length$C_{ref} = 7.00532\ m$. Table 5 describes the free-stream flow conditions. A coarse mesh of approximately 4 million cells is used. The first grid node from the wall is located at $y^+ \approx 1$. The SpalartAllmaras (SA) turbulence model is used for this simulation. The flow angle for this mesh to gain $C_L = 0.5$ is $\alpha = 2.36^o$.

M	0.85
P_{total}	201326.91 Pa
T_{total}	310.93 K

Table 5: NASA CRM free-stream conditions.

The pressure coefficients at locations 13.06%, 28.3% and 50.24% of the wing span is shown in the Figure 36. The results are compared with the wind tunnel data at these locations. This experimental data is for C_L values of 0.486 and 0.52 as shown in [81]. A good level of agreement with the measurements can be observed in these plots. The pressure distribution on the aircraft is shown in the Figure 36(d).

4. Conclusions and future work

In the first part of the paper, various automatic and manual blocking methods are compared in the first part of this paper. The adjoint error es-

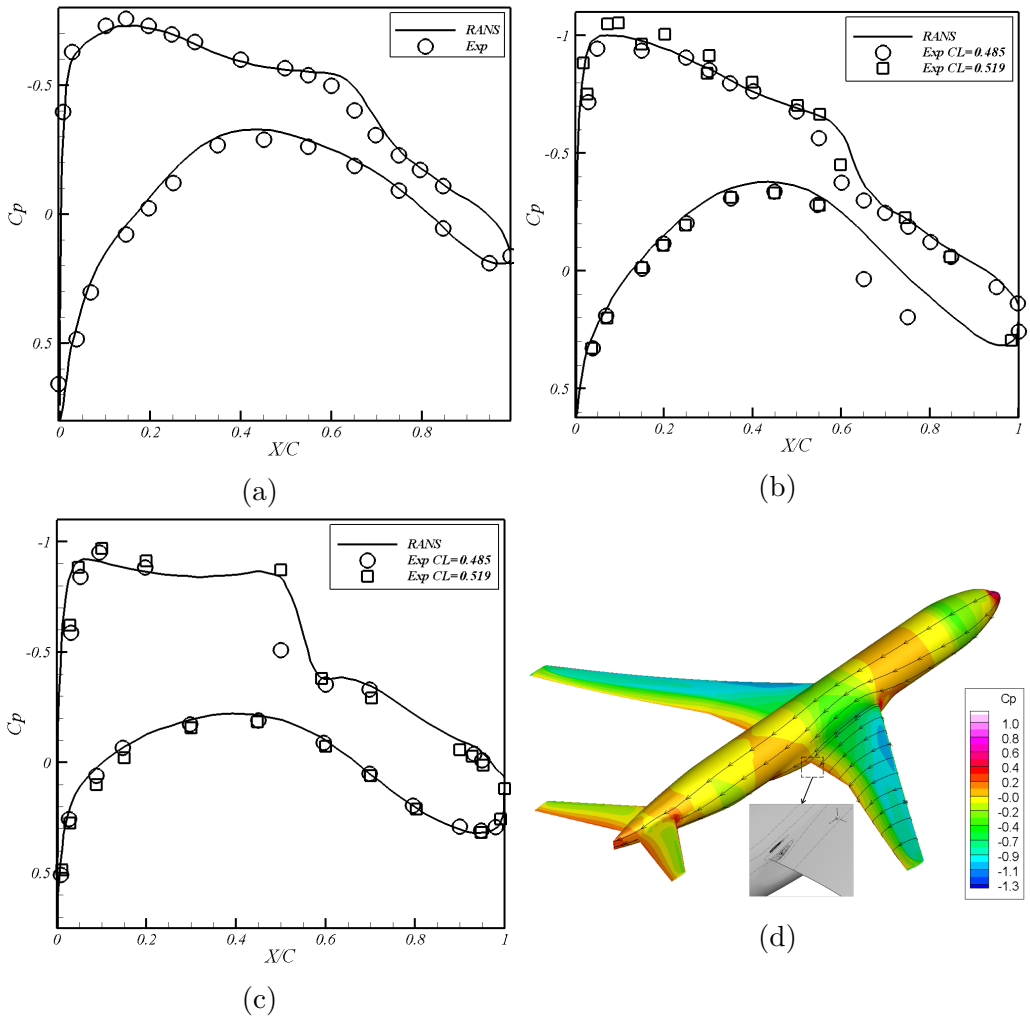


Figure 36: NASA CRM wing-body-tail pressure coefficients at (a) 13.06% (b) 28.3% and 50.24% (c) of the wing span (d) C_p distribution.

timation performed on the meshes generated over various block topologies shows that, in general, medial axis based methods produce optimal domain decomposition. This is due to the fact that a better flow alignment and a more uniform cell size distribution is produced by these techniques. Such a technique is useful for many internal flows where the inlet flow angle remains constant. The cross field blocking performs better when it results in a block topology similar to the one generated by the medial axis methods. We have also observed that the manually generated block topologies yield better results in some cases while not performing the best in the others. This can be attributed to the experience of the user. Thus the automated approaches can guide a CFD practitioner to optimally block the domain with the advantage of being less user intensive.

In the second part of the paper, a novel hybrid blocking approach is demonstrated for the cases where previous techniques might not perform well. The approach needs further development but is useful keeping in the view that a fully automated 3D blocking method has yet to be proposed. The technique is general and can be applied to a broad range of flow domains other than the aerodynamics cases. Such approaches can also be employed to produce standard templates for various geometric configurations.

There are many aspects of the analysis yet to be considered. Flow aligned meshes provide better accuracy. Hence, an objective function based on the flow angle can be used to obtain the adjoint sensitivity. In this way, the mesh can be adapted to have better flow alignment. Also, using a combination of the objective functions for the adjoint based analysis could be more useful. Work is under way to automate and extend the range of applicability of the hybrid blocking approach. Also, block boundary and the distance field iso-surface adaptation based upon adjoint error estimate is being developed. Finally, all these methods should be considered as few of the many options towards automatic mesh generation.

Acknowledgment

The authors would like to thank the Higher Education Commission, Pakistan and the Rolls-Royce plc for the funding and permission to publish this work. Authors would also like to thank David Radford and Dr. Caleb Dhanasekaran for providing the test cases, technical support and for the useful discussions. The comments and suggestions by Dr. James Tyacke are also appreciated.

Appendix

Error Estimation

Consider a computational domain Ω and let Ω_H and Ω_h are the coarse mesh and the fine mesh discretizations of this domain respectively. H and h ($H > h$) here represent the mesh length scales associated with a particular discretization such as finite difference or finite volume. The coarse mesh has a mesh density that is able to capture the basic feature of the flow but might not yield the desired level of accuracy. The fine mesh on the other hand is a systematic uniform refinement of the coarse mesh which can provide the accuracy required but is computationally expensive. The discretization of the governing equations on the coarse and fine mesh yield the residual vectors which can be denoted by $R_H(Q_H)$ and $R_h(Q_h)$ respectively where Q is the solution of system of the governing flow equations. Let $J(Q)$ be the objective function which one wants to estimate. The estimation of this objective function on the coarse and fine mesh are denoted by $J_H(Q_H)$ and $J_h(Q_h)$. An estimate of $J_h(Q_h)$ on the fine grid, without solving on the fine grid, can be made by a Taylor's series expansion of $J_h(Q_h)$ about the solution on the coarse grid

$$J_h(Q_h) = J_h\left(Q_h^H\right) + \left.\frac{\partial J_h}{\partial Q_h}\right|_{Q_h^H}\left(Q_h - Q_h^H\right) + ... \tag{9}$$

Also the expansion of $R_h(Q_h)$ about the coarse mesh yields

$$R_h(Q_h) = R_h\left(Q_h^H\right) + \left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H}\left(Q_h - Q_h^H\right) + ... \tag{10}$$

The vector $\left(Q_h^H\right)$ in the above equation is the coarse mesh solution estimated on the fine mesh with a suitably defined prolongation operator. The vector $\left.\frac{\partial J_h}{\partial Q_h}\right|_{Q_h^H}$ in Eqn. (9) represents the linear sensitivity of the fine mesh function with respect to $\left(Q_h^H\right)$. Moreover, the vector $\left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H}$ in Eqn. (10) is the fine mesh Jacobian evaluated using the projected coarse mesh solution. $\left(Q_h^H\right)$ can be evaluated by

$$Q_h^H = I_h^H Q_H \tag{11}$$

where I_h^H represents an appropriate prolongation operator. For example the coarse grid solution can be reconstructed on the fine mesh using linear or higher order interpolation. Assuming the well-posedness, the Eqn. (10) can be inverted. Also knowing that $R_h(Q_h) = 0$ for steady state problem, Eqn. (10) yields

$$\left(Q_h - Q_h^H\right) \approx -\left\{ \left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H} \right\}^{-1} R_h\left(Q_h^H\right) \tag{12}$$

From Equations (12) and (9) we get

$$J_h(Q_h) \approx J_h\left(Q_h^H\right) - (\nu_h|_{Q_h^H})^T R_h\left(Q_h^H\right) \tag{13}$$

where $\nu_h|_{Q_h^H}$ is the discrete adjoint solution vector estimated at the fine mesh using Q_h^H . This adjoint solution vector satisfies the equation

$$\left\{ \left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H} \right\}^T \nu_h|_{Q_h^H} = \left\{ \left.\frac{\partial J_h}{\partial Q_h}\right|_{Q_h^H} \right\}^T \tag{14}$$

Eqn. (13) requires the evaluation of the term $(\nu_h|_{Q_h^H})^T$ on the fine grid. To avoid this evaluation, $(\nu_h|_{Q_h^H})^T$ can also be estimated thorough coarse grid adjoint interpolation onto the fine grid via some projection operator

$$\nu_h^H = I_h^H \nu_H \tag{15}$$

where the coarse mesh adjoint solution ν_H is the solution of following adjoint equations on the coarse mesh

$$\left\{ \frac{\partial R_H}{\partial Q_H} \right\}^T \nu_H = \left\{ \frac{\partial J_H}{\partial Q_H} \right\}^T \tag{16}$$

The final estimate of the objective function $J(Q)$ is then given by

$$J(Q) = J_h\left(Q_h^H\right) - (\nu_h^H)^T R_h\left(Q_h^H\right) \tag{17}$$

The second expression on the right hand side of the above equation is called the error correction term. It can be seen that the error in the objective function is related to the local residual error through the adjoint variables which act as weight function. Eqn. (13) can be disintegrated into the following

form

$$J_h\left(Q_h^H\right) - J_h\left(Q_h\right) \approx \left(\nu_h^H\right)^T R_h\left(Q_h^H\right) + (\nu_h|_{Q_h^H} - \nu_h^H)^T R_h\left(Q_h^H\right) \qquad (18)$$

The first term on the right hand side of Eqn. (18) is the main computable error estimate while the second term is the error in this computable error estimate. This first term on the right hand side shows that the adjoint variables directly relates the error in the given functional to the local residual errors. Stating in another way, the adjoint solution act as a weight function to the local residual error and gives the effect of the residual error on the output functional.

References

[1] P. G. Tucker, Unsteady Computational Fluid Dynamics in Aeronautics, Springer, 2013.

[2] S. Shahpar, L. Lapworth, PADRAM: Parametric design and rapid meshing system for turbomachinery optimisation, in: ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference, American Society of Mechanical Engineers, 2003, pp. 579–590, ASME Paper No. GT 2003-38698.

[3] A. Milli, S. Shahpar, PADRAM: Parametric design and rapid meshing system for complex turbomachinery configurations, in: ASME Turbo Expo 2012: Turbine Technical Conference and Exposition, American Society of Mechanical Engineers, 2012, pp. 2135–2148, ASME Paper No. GT 2012-69030.

[4] H. Blum, A transformation for extracting new descriptions of shape, Models for the Perception of Speech and Visual Form (1967) 362380.

[5] J.-D. Boissonnat, Geometric structures for three-dimensional shape representation, ACM Transactions on Graphics (TOG) 3 (4) (1984) 266–286.

[6] N. Amenta, S. Choi, R. Kolluri, The power crust, in: Proceedings of the sixth ACM symposium on Solid modeling and applications, ACM, 2001, pp. 249–266.

[7] T. K. Dey, W. Zhao, Approximate medial axis as a Voronoi subcomplex, Computer-Aided Design 36 (2) (2004) 195–202.

[8] T. K. Dey, W. Zhao, Approximating the medial axis from the voronoi diagram with a convergence guarantee, Algorithmica 38 (1) (2004) 179–200.

[9] K. Palágyi, A 3D fully parallel surface-thinning algorithm, Theoretical Computer Science 406 (1) (2008) 119–135.

[10] Z. Guo, R. W. Hall, Parallel thinning with two-subiteration algorithms, Communications of the ACM 32 (3) (1989) 359–373.

[11] L. Lam, S.-W. Lee, C. Suen, Thinning methodologies-a comprehensive survey, IEEE Transactions on pattern analysis and machine intelligence 14 (9) (1992) 869–885.

[12] Y. Zhang, P. Wang, S.-P. Patrick, Analytical comparison of thinning algorithms, International journal of pattern recognition and artificial intelligence 7 (05) (1993) 1227–1246.

[13] P.-E. Danielsson, Euclidean distance mapping, Computer Graphics and image processing 14 (3) (1980) 227–248.

[14] I. Ragnemalm, The Euclidean distance transform in arbitrary dimensions, Pattern Recognition Letters 14 (11) (1993) 883–888.

[15] J. Vleugels, M. Overmars, Approximating generalized voronoi diagrams in any dimensionTechnical report UU-CS-1995-14 Utretcht University.

[16] H. Xia, P. Tucker, Finite volume distance field and its application to medial axis transforms, International Journal for Numerical Methods in Engineering 82(1) (2010) 114134.

[17] M. Etzion, A. Rappoport, Computing Voronoi skeletons of a 3-D polyhedron by space subdivision, Computational Geometry 21 (3) (2002) 87–120.

[18] T. Culver, J. Keyser, D. Manocha, Accurate computation of the medial axis of a polyhedron, in: Proceedings of the fifth ACM symposium on Solid modeling and applications, ACM, 1999, pp. 179–190.

[19] V. Milenkovic, Robust construction of the voronoi diagram of a polyhedron., in: CCCG, Vol. 93, Citeseer, 1993, pp. 473–478.

[20] E. C. Sherbrooke, N. Patrikalakis, E. Brisson, Computation of the medial axis transform of 3-D polyhedra, in: Proceedings of the third ACM symposium on Solid modeling and applications, ACM, 1995, pp. 187–200.

[21] J. M. Reddy, G. M. Turkiyyah, Computation of 3D skeletons using a generalized delaunay triangulation technique, Computer-Aided Design 27 (9) (1995) 677–694.

[22] T. Tam, C. Armstrong, 2D finite element mesh generation by medial axis subdivision, Adv. Eng. Software 13 (1991) 313324.

[23] M. Price, C. Armstrong, Hexahedral mesh generation by medial surface subdivision: Part I. Solids with Convex Edges, International Journal for Numerical Methods in Engineering 38 (1995) 33353359.

[24] M. Price, C. Armstrong, Hexahedral mesh generation by medial surface subdivision: Part II. Solids with Flat and Concave Edges, International Journal for Numerical Methods in Engineering 38 (1995) 33353359.

[25] D. Sheehy, C. Armstrong, D. Robinson, Computing the medial surface of a solid from a domain delauney triangulation, in: ACM Symposium on Solid Modeling Foundations and Applications, 1995, pp. 201–212.

[26] D. Rigby, : A technique for automatic multi-block topology generation using the medial axis, NASA/CR FEDSM2003-45527.

[27] W. R. Quadros, LayTracks3D: a new approach to meshing general solids using medial axis transform, Procedia Engineering 82 (2014) 72–87.

[28] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49.

[29] J. A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Vol. 3, Cambridge university press, 1999.

[30] P. G. Tucker, Differential equation-based wall distance computation for DES and RANS, Journal of Computational Physics 190 (1) (2003) 229–248.

[31] P. G. Tucker, C. L. Rumsey, P. R. Spalart, R. Bartels, R. Biedron, Computations of wall distances based on differential equations, AIAA journal 43 (3) (2005) 539–549.

[32] E. Fares, W. Schröder, A differential equation for approximate wall distance, International journal for numerical methods in fluids 39 (8) (2002) 743–762.

[33] P. Tucker, Differential equation-based wall distance computation for DES and RANS, Journal of Computaional Physics 190(1) (2003) 229248.

[34] H. Xia, P. Tucker, Fast equal and biased distance fields for medial axis transform with meshing in mind, Applied Mathematical Modelling 35 (2011) 580419.

[35] D. Bommes, T. Lempfer, L. Kobbelt, Global structure optimization of quadrilateral meshes, in: Computer Graphics Forum, Vol. 30, Wiley Online Library, 2011, pp. 375–384.

[36] Y. L. i, W. Wang, R. Ling, C. Tu, Shape optimization of quad mesh elements, Computers & Graphics 35 (3) (2011) 444–451.

[37] J. Palacios, E. Zhang, Rotational symmetry field design on surfaces, in: ACM Transactions on Graphics (TOG), Vol. 26, ACM, 2007, p. 55.

[38] N. Kowalski, F. Ledoux, P. Frey, A PDE based approach to multidomain partitioning and quadrilateral meshing, in: Proceedings of the 21st international meshing roundtable, Springer, 2013, pp. 137–154.

[39] H. J. Fogg, C. G. Armstrong, T. T. Robinson, Automatic generation of multiblock decompositions of surfaces, International Journal for Numerical Methods in Engineering 101 (13) (2015) 965–991.

[40] H. J. Fogg, C. G. Armstrong, T. T. Robinson, Enhanced medial-axis-based block-structured meshing in 2-D, Computer-Aided Design.

[41] J. Huang, Y. Tong, H. Wei, H. Bao, Boundary aligned smooth 3D cross-frame field, in: ACM Transactions on Graphics (TOG), Vol. 30, ACM, 2011, p. 143.

[42] Y. Li, Y. Liu, W. Xu, W. Wang, B. Guo, All-hex meshing using singularity-restricted field, ACM Transactions on Graphics (TOG) 31 (6) (2012) 177.

[43] N. Kowalski, F. Ledoux, P. Frey, Block-structured hexahedral meshes for cad models using 3d frame fields, Procedia Engineering 82 (2014) 59–71.

[44] P. Alliez, D. D. D. Cohen-Steiner, O. Devillers, B. Lévy, M. Desbrun, Anisotropic polygonal remeshing, in: ACM Transactions on Graphics (TOG), Vol. 22, ACM, 2003, pp. 485–493.

[45] F. Kälberer, M. Nieser, K. Polthier, QuadCover-Surface parameterization using branched coverings, in: Computer Graphics Forum, Vol. 26, Wiley Online Library, 2007, pp. 375–384.

[46] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, H. Bao, Spectral quadrangulation with orientation and alignment control, in: ACM Transactions on Graphics (TOG), Vol. 27, ACM, 2008, p. 147.

[47] I. Malcevic, Automated blocking for structured CFD gridding with an application to turbomachinery secondary flows, in: 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, 2011.

[48] C. J. Roy, Review of discretization error estimators in scientific computing, AIAA 2010–126.

[49] T. J. Baker, Mesh adaptation strategies for problems in fluid dynamics, Finite Elements in Analysis and Design 25 (3) (1997) 243–273.

[50] L. Baskett, R. Haimes, Feature extraction of shear layers, in: AIAA Computational Fluid Dynamics Conference, 15 th, Anaheim, CA, 2001.

[51] S. Pirzadeh, An adaptive unstructured grid method by grid subdivision, local remeshing, and grid movement, AIAA paperAIAA Paper No. 99-3255.

[52] F. M. White, I. Corfield, Viscous fluid flow, Vol. 3, McGraw-Hill New York, 2006.

[53] C. Roy, Strategies for driving mesh adaptation in CFD, AIAA Paper 2009–1302.

[54] R. P. Dwigh, Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation, Journal of Computational Physics 227 (5) (2008) 2845–2863.

[55] D. Venditti, D. Darmofal, Grid adaptation for functional outputs: Application to two-dimensional inviscid flow, Journal of Computational Physics 176 (2002) 40–69.

[56] G. P. Warren, W. K. Anderson, J. L. Thomas, S. L. Krist, Grid convergence for adaptive methods, AIAA paperAIAA Paper No. 91-1592.

[57] J. Muller, M. Giles, Solution adaptive mesh refinement using adjoint error analysis, AIAA Paper 2001-2550.

[58] M. B. Giles, N. A. Pierce, Adjoint error correction for integral outputs, Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics, Springer 25.

[59] M. Park, Adjoint-based, three dimensional error prediction and grid adaptation, AIAA Journal 42 (2004) 18541862.

[60] R. Dwight, Goal-oriented mesh adaptation using a dissipation-based error indicator, in: Proceedings of ICFD Conference, University of Reading, 2007.

[61] K. Fidkowski, A simplex cut-cell adaptive method for high-order discretizations of the compressible NavierStokes equations, Ph.D. thesis, Massachusetts Institute of Technology (2007).

[62] M. Nemec, M. J. Aftosmis, Error estimation and adaptive refinement for embedded-boundary Cartesian meshes, AIAA Paper 2007–4187.

[63] K. Mani, D. J. Mavriplis, Discrete adjoint based time-step adaptation and error reduction in unsteady flow problems, AIAA Paper 2007–3944.

[64] K. Fidkowski, D. Darmofal, Review of output-based error estimation and mesh adaptation in Computational Fluid Dynamics, AIAA Journal 49(4) (2011) 673–694.

[65] M. B. Giles, N. A. Pierce, Adjoint error correction for integral outputs, Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics, Springer 25.

[66] D. A. Venditti, D. L. Darmofal, Grid adaptation for functional outputs: Application to two-dimensional inviscid flows, Journal of Computational Physics 176 (1) (2002) 40–69.

[67] L. Lapworth, Hydra-CFD: a framework for collaborative CFD development, in: International Conference on Scientific and Engineering Computation (IC-SEC), Singapore, June, Vol. 30, 2004.

[68] P. Moinier, Algorithm developments for an unstructured viscous flow solver, Ph.D. thesis, Oxford University (1999).

[69] M. B. Giles, M. C. Duta, J.-D. Müller, N. A. Pierce, Algorithm developments for discrete adjoint methods, AIAA journal 41 (2) (2003) 198–205.

[70] M. C. Duta, S. Shahpar, M. B. Giles, Turbomachinery design optimization using automatic differentiated adjoint code, in: ASME Turbo Expo 2007: Power for Land, Sea, and Air, American Society of Mechanical Engineers, 2007, pp. 1435–1444, ASME Paper No. GT 2007-28329.

[71] H. Xia, P. G. Tucker, Finite volume distance field and its application to medial axis transforms, International Journal for Numerical Methods in Engineering 82 (1) (2010) 114–134.

[72] D. L. Rigby, Topmaker: A technique for automatic multi-block topology generation using the medial axis, NASA/CR 213044.

[73] I. Malcevic, Automated blocking for structured CFD gridding with an application to turbomachinery secondary flows, in: 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, 2011, aIAA Paper No. AIAA 2011-3049.

[74] H. J. Fogg, C. G. Armstrong, T. T. Robinson, New techniques for enhanced medial axis based decompositions in 2-D, Procedia Engineering 82 (2014) 162–174.

[75] Pointwise, `http://www.pointwise.com`.

[76] P. R. Spalart, A one-equation turbulence model for aerodynamic flows, Recherche Aerospatiale 1 (1994) 5–21.

[77] U. R. Oriji, Numerical investigation of intake flows in crosswinds, Ph.D. thesis, Department of Engineering, University of Cambridge (2014).

[78] M. Judge, T. Hynes, Experimental investigation into effect of roughness on crosswind separation Report, Whittle Lab, University of Cambridge, UK (2011) 1–81.

[79] J. C. Vassberg, M. A. DeHaan, S. M. Rivers, R. A. Wahls, Development of a common research model for applied CFD validation studiesAIAA Paper No. AIAA-2008-6919.

[80] J. Vassberg, E. N. Tinoco, M. Mani, B. Rider, T. Zickuhr, D. Levy, O. P. Brodersen, B. Eisfeld, S. Crippa, R. A. Wahls, et al., Summary of the Fourth AIAA CFD Drag Prediction WorkshopAIAA Paper No. AIAA-2010-4547.

[81] D. W. Levy, K. R. Laflin, E. N. Tinoco, J. C. Vassberg, M. Mani, B. Rider, C. Rumsey, R. Wahls, J. H. Morrison, O. P. Brodersen, et al., Summary of data from the fifth AIAA CFD drag prediction workshopAIAA Paper No. AIAA-2013-0046.

